Chapter 1. Basic Concepts

Table of Contents

Introduction ................................................................. 1
Hypertext ........................................................................... 2
Anchors and Links .......................................................... 2
Jumps ................................................................................ 3
Chain of Links ................................................................... 3
Loops and Mesh ............................................................... 4
Hypermedia ................................................................. 4
Authoring Hypertxt .......................................................... 4
The Ultimate Hypermedia System: The World Wide Web ................................................ 5
Basic Ideas of the Web ................................................... 6
Summary of Web Terminologies ....................................... 7
Web Servers ................................................................. 14
Distributed Processing ................................................... 14
Review Questions ........................................................... 16
Review Question 1 ............................................................ 16
Review Question 2 ............................................................ 16
Review Question 3 ............................................................ 16
Review Question 4 ............................................................ 17
Review Question 5 ............................................................ 17
Review Question 6 ............................................................ 17
Answers ........................................................................... 17
Discussions for Exercise 1.2 ............................................. 17
Answers to Exercise 1.3 .................................................... 18
Answers to Exercise 1.4 .................................................... 18
Answers to Exercise 1.5 .................................................... 19
Answers to Exercise 1.6 .................................................... 19
Answers to Review Question 1.1 ....................................... 19
Answers to Review Question 1.2 ....................................... 19
Answers to Review Question 1.3 ....................................... 19
Answers to Review Question 1.4 ....................................... 19
Answers to Review Question 1.5 ....................................... 20
Answers to Review Question 1.6 ....................................... 20

Introduction

The Internet is just over 30 years old and the World Wide Web has passed the 10 year mark. Furthermore, the concepts of computer networks and hypertext on which these technologies rely are only a little older. And yet the speed of development of these technologies, the speed of uptake by companies, and the speed of acceptance by consumers is unlike anything mankind has witnessed. Although both the Internet and the Web are firmly rooted in academic, altruistic endeavour, there is no doubt that the hoped-for opportunities for business are driving much of the development, change and improvement. This module aims to prepare you for contributing to this endeavour by assisting you to understand the basic ideas and technologies and by giving you the opportunity to make abstract ideas concrete by designing and writing Web pages using HTML and JavaScript.

The module starts here with, inevitably, the more theoretical aspects of the Internet and the Web. We start with the notion of hypertext before moving on to the most elaborate hypermedia system, the Web,
and the ideas of client–server computing that allow us to use it.

**Hypertext**

Take a dictionary and observe how its content is linked together. How do you search for the meaning of a word? How can you find another word synonymous with that word? The dictionary is a paper example of a hypertext system. So are encyclopaedias, product catalogues, user help books, technical documentation and many others. Information is obtained by searching through some kind of indexing - the dictionary is arranged in alphabetical order. Readers are then pointed to the page of any other related information. They can read the information they are interested in without having to read the document sequentially from page one.

Hypertext systems allow for non-sequential, or non-linear, reading. That is the underlying idea of a hypertext system. The result is a multidimensional document that can be explored by following different paths through it. In this section, we will look into the application of hypertext in computer systems, mainly the World Wide Web hypertext system.

The main use of hypertext is in information retrieval applications. The ease of linking different pieces (fragments) of information is the important aspect of hypertext information retrieval. The type of information can be of various media; it may be fragments of textual documents, structured data from databases, or list of terms and their definitions, making up the contents of a document.

Therefore, in a hypertext system it is possible to:

- link with a term that represents aspects of the content of a document
- connect two related documents
- relate a term to a fragment containing its definition and use
- link two related terms

Such a hypertext system becomes a storage of a large collection of textual and multimedia documents. This automatic construction of such a database gives the end-user access to a large depository of knowledge for reading, browsing and retrieving to satisfy their information needs. This is the reason why such hypertext system is called a digital library. The Web started as an extensively large digital library. As it has grown in popularity, it has offered the possibility of commerce on the Internet, making it much more than a digital library.

**To do:**

Read about networked hypertext and hypermedia in your textbooks.

**Anchors and Links**

A hypertext document is one that contains links referring to another part of the document, or some other document or documents. A hypertext document does not have to be read serially; the fragments of information can be accessed directly.

The reference links embedded in a document are known as hyperlinks, which, when selected, will cause another fragment of the document to be displayed. The reader of a document may jump to another part of the same page, another page in the same document or another document via the link at the anchor typically by clicking on the anchor. By following a series of hyperlinks, he or she could implicitly follow a chained path with a series of jumps.

A computerised hypertext system implements this idea by including anchors and links in documents,
which are usually represented by files. An anchor is a fragment of information to which a link to 
typically another document may be attached. A link is simply a reference (or a pointer). For example, 
in the diagram below, the fragment of Document A containing ‘You can find this in Section 5 of B’ is an 
anchor from which there is a link to the relevant section in Document B.

Take care not to confuse anchor or link. (Many non-computer people do when using a Web browser.) A 
link is a pointer to another piece of information within the same document or in another document; often 
you cannot see how that link is established (it may be a hidden URL or some other programmed mech-
anism). An anchor is a fragment of information to which a link is attached; in a Web browser it is what 
you see. Take the following link as an example. The phrase ‘Back to top’ in the line below is an anchor. 
It contains the link to top of this page.

A hyperlink must have unambiguous reference to the document (an identifier like a file name), informa-
tion on its location (where in some file space or network it is) and the mechanisms to access it (how to 
traverse). In Unit 2 you will meet HTML anchors and how the referenced documents are identified and 
located with URLs.

Jumps

A hypertext document allows loops to previous fragments of the documents. The resulting topology 
would then be a mesh network of fragments of information.

The table of contents in a book is a collection of anchors with explicit links to the internal parts of the 
book. Its bibliography is another collection of links but it refers to external information. To refer to the 
internal parts of the book is simple, the reader merely turns to the appropriate page, usually identified by 
page number. However, referring to the external information given in a bibliography requires a more 
complicated effort of searching.

In computer-based hypertext documents, the mechanism to follow a pointer (the jump) is automatic. 
Jumping to an external link (another document) is as easy as jumping to an internal link within the same 
document. As long as the link is sufficiently specified with the name and the exact location of the linked 
document, the user can access directly the linked document with a simple click on the anchor.

Chain of Links
A series of successive jumps implicitly constructs a chained path. There is no limit as to the number of jumps, therefore the size of the chain is not constrained.

There may be more than one link in a page and the reader is free to choose any of these links to follow. The path a reader takes will then be different from the path of another reader. Each sequence of jumps forms a different path to fragments of the overall information in the hypertext document. Generally, there is no rigid order to read the information.

There are two different but complementary purposes of chaining documents:

- **Focusing**: At each step of the path, the user narrows the scope of the search until the fragment containing the topic of interest is reached.

- **Broadening**: This comes from chained links between documents containing multiple outgoing pointers. This is useful when the user does not have a precise idea of what is being looked for, or wishes to conduct a broad search in a certain domain.

Going through paths of hypertext documents usually poses no technical difficulty. However, you can expect practical difficulties in retrieving a particular piece of information from a path with numerous alternative links.

**Loops and Mesh**

As the reader is free to choose which links therefore the paths to take, it is possible for a user to return to where he or she started or to some point previously visited. In other words, loops may exist. A path may even return to the original (home) document. Hence, the structure does not follow a linear pattern; instead, a mesh network is created by the path.

This critical property shifts the burden of devising suitable exploration paths from the designer of a system of documents to the users. This changes the way information is stored and retrieved. Instead of searching directly for information, hypertext allows browsing for information. However, the mesh of information creates difficulty in navigating through the inextricable network.

**Hypermedia**

One of the original purposes for hypertext is for the storage and management of textual documents. As the technology in computers and telecommunications improved, the capabilities of hypertext systems have extended to include any kind of media that can be digitised, such as sound and images.

Therefore, music and videos can be accessed via hyperlinks. This blending of hypertext and multimedia is known as Hypermedia. A combination of text, graphics, video or sound can now easily be interlinked in hypermedia document to offer a rich, often interactive, environment.

**Authoring Hypertext**

The process of preparing hypertext documents or, quite often, of converting a flat (linear) collection of documents to hypertext, is referred to as authoring.

Often an initial collection of documents has to be reorganised by splitting up the original documents. Then links between the new documents must be constructed. Authors of hypertext documents are not only responsible for their content, but must link documents together, create paths through related documents, and built references that point to external documents associated with it.

Conceptually, related information is ultimately presented as a single, unique collection of hypertext documents. The remarkable aspect of a hypertext or hypermedia documents that distinguishes it from any
other types of document is that the unique collection is ‘shaped’ by the user as he or she navigates the hypertext network. Each sequence of the links is a possible exploration path and each sequence chosen forms a single conceptual document for the user.

The hypertext document therefore becomes multi-dimensional with the various possible exploration paths. Each dimension is dynamically activated as different paths are taken.

### Table 1.1.

| Getting lost in ‘hyperspace’ | The easy linking of different fragments of information, crucial for browsing, can produce hypertext documents that are very difficult to use. The user may become disoriented when they do not know where they are and where he can go to. This problem of navigating a hypertext network is also known as ‘lost in hyperspace’. There are ways to minimise the risks of being lost in such a large information space. |
| Return path | The user simply reverses the chain, backtracking through all the previous documents till he or she reaches the one they want to read again. Alternatively, if the user remembers the reference of the required document, it may be selected from a list of the most recent documents explored. |
| Home page | The starting fragment in a path is known as the home page. This home page is usually a well-defined document that contains the first links to a certain path. It helps to remind the user the path he has taken before and may even serve as a starting point to another path. |
| Overview diagrams | This is the explicit display of the meshed network of documents and links. Many Web sites have an overview site map showing the paths the user may take to access certain information from the site. |
| Guided tours | These are suggested paths arranged by the authors of the documents. Its purpose is to assist the user in the exploration for information in the network. Tour documents form a logical path sequence by using simple ‘next-document’ or ‘last-document’ anchors. |
| Direct jump | The user has to know the name and location of the documents to be connected directly to the documents. The URL address of the Web site is typed in and the requested page retrieved and displayed to the user. |
| Content-based retrieval | Some documents may offer a search facility. Browsing for information through the search facility can help narrow the information space to the domain of interest. However, most current search facilities are restricted to textual information only. |

### The Ultimate Hypermedia System: The World Wide Web
Basic Ideas of the Web

The World Wide Web (Web) implements hypermedia. It has largely achieved the goal of Tim Berners-Lee, its British inventor, of a universal information space. Thanks to the global reach of the Internet, there is potentially universal access to an enormous volume of documents over the Internet. (Of course, in many developing countries, access is poor, which raises issues of disenfranchisement and disempowerment.) Many organisations make publicly available collections of hypermedia documents as part of their marketing programme, customer service and global operations. Computer suppliers, for example, now publish very detailed specifications of their products via the Web.

Web servers and clients may be located at any part of the world, connected by telecommunication links. If the Web is in some sense a digital library, it is one with no geographical location. When it comes to commerce, distance begins to lose importance. As long as a supplier can provide goods or services where they are required, the location of the vendor and the consumer will not matter. (This gives rise to issues about jurisdiction for taxes, consumer laws, legality of product, etc.) This absence of distance is supported by the ease with which Web documents may be located world-wide; the mechanism is straightforward thanks to the way the location of such 'resources; are identified by a Uniform Resource Location (URL). The URL format unambiguously specifies locations of 'documents' (document is attaining a new meaning) on the Web. These document addresses point to where and how to find a document. This location mechanism allows the actual implementation of geography-independent feature of the Web.

Of course generally speaking, there is no central authority controlling the Web, although fully qualified domain names are subject to controlled allocation, and Internet Service Providers may be subject to the laws of the countries in which they operate. Furthermore, the World-Wide Web Consortium (W3C), headed by Tim Berners-Lee at Massachusetts Institute of Technology, is influencing, and to a large degree controlling how technologies are deployed on the Web. The W3C specifies HTML and XML, but others, like ECMA, the European Computer Manufacturers Association, have standardised what we mostly call JavaScript, which was actually developed by Netscape.

Anyone who knows how and has access to server space, can create a Web document, make reference to any other document and locate a document where a server can supply it. Moreover, a user does not require specifically authorised software on their computer platform to access the Web. Many Web browsers are free; all browsers can access the information on the Web, but not all can interact with Web pages. For example, if Java applets are prohibited or a browser does not support JavaScript, much modern information will not be accessible, especially when software generates what might have been document content.

The implications are easy to predict. Chaos might be expected, as can be seen from the mesh intricacies of the Web and the expanding difficulties of navigating through the information space. However, even the most inexperienced users cope and the Web is becoming a truly universal world of information, and hence a universal place for doing business.

A striking feature of the web, which has guaranteed its success, is the way in which user interfaces have a high degree of commonality leading to ease of use. In the beginning of the Web, for a year or two, most Web pages looked the same and there was little interactivity. But browsers are similar and generally attractive and usable designs have encouraged the emergence of commerce on the web. In fact, incompatibilities in data representation between the local (client) system and the remote servers, and how the information is brought to the local system, are made minor by the common languages for displaying documents and generating user interfaces. It is basically the logical structure of Web documents expressed in the Hypertext Markup Language (HTML) that results in document layout, so structure helps smooth differences.

Furthermore, dynamic pages can respond interactively to user input. It is possible to execute a program to create a page as it is requested. Web pages are increasingly being used as a front end to databases.

The user can fill out a query and send it off for processing. The server queries to the database and returns the output in HTML. To allow data to be sent to and from servers, a standard called Common
Gateway Interface (CGI) was created. The browser will take care of masking any format and access method differences.

It is then possible to embed programs inside HTML. When the browser loads such a page, the code is immediately executed. This mechanism supports remote transactions in the commercial aspect of the Web

To Do:
Read about the World-Wide Web in your textbooks.

Summary of Web Terminologies

Here we briefly summarise some of the terms you will need early in the module but which will be studied to varying extents later

Protocols and Network

A network protocol is a standard way of regulating data transmission between computers. Just as diplomats adhere to protocols, rules of behavior, when in foreign lands, network communications do the same – they have to obey agreed rules if they are to ‘get on with each other’. After many years of both public and private research and development of networks, one unlikely candidate now dominates the world – in fact a pair of protocols TCP (Transaction Control Protocol) and IP (Internet Protocol), together known as TCP/IP. (It was unlikely because faster, standardized protocols had been agreed but none had the same robustness and extensibility as TCP/IP, and none had the Web developed using them.) The ubiquity of TCP/IP is why business is so keen to exploit the Internet, and hence the Web.

Very often protocols were implemented without any formal acceptance and, because they worked most of the time, became standards by default. Although TCP/IP is an accepted, de facto standard, work on Internet protocols continue so that growth can continue and quality improve. There is no dictating authority for the Internet. Without a controlling authority, interim proposals about protocol changes are taken by groups of interested individuals, and then opened up for discussion. Documents containing the various proposed standards were published as Requests For Comment documents (RFCs). You may see references to a specific RFC as the best description of a protocol!

Uniform Resource Locator (URL)

To locate any resources on the web, a URL is needed. It is an address format that specifies how and where to get a document. The general format is as follows, where the various items in italics must be substituted with part of a real URL, or omitted.

```
http://machine_name:port/path/file_name.file_extension
```

- **machine_name** is either an IP address, for example 137.234.33.89 or Fully Qualified Domain Name (also known as a DNS name, because Domain Name Servers need to know them), for example, www.apple.com [http://www.apple.com]
- **port** is the TCP port to connect to; this is an entry point to software on the server; an optional part of a URL
- **path** is a relative file path from the server’s document root; the server will start looking for a file in a specific directory and paths are relative to this
- **file_name** is the name of the file to be browsed, e.g. welcome
file_extension is one of a number of suffixes which by convention and operating system setup indicate the type of data contained within the file, e.g. htm,html, txt.

**HyperText Markup Language (HTML)**

This language provides the format for specifying simple logical structures and the hypertext links. As a mark-up language, special formatting commands must be placed in the body of text describing how the final version would appear. These formatted documents need to be compiled by the browser which uses the HTML code to format the page being displayed. Several units in this module deal with HTML extensively. Although most professionals use special authoring tools to write HTML documents and to manage sites, developers of e-commerce sites and applications need to know the nitty-gritty detail, and this is what you will study.

**HyperText Transfer Protocol (HTTP)**

HTTP is a protocol retrieving documents from a variety of machines in a minimum of time. It was invented by Tim Berners-Lee to support a project in developing a distributed hypertext system. Distributed hypertext requires the retrieval of documents from many different machines. File Transfer Protocol (FTP) which predates the Web, would be too slow for this purpose as it is a protocol that requires a client to connect to a server and there would be a connection establishment and maintenance overhead when accessing different machines.

Therefore, to support browsing, HTTP has the following characteristics:

- connection-less: a connection is established only for the period of transfer;
- stateless: the server has no ‘history’ of client visits (although the implementation of cookies overcomes this);
- comprehensive addressing: diverse files on any HTTP server world-wide can be referenced
- diverse data: using extendable MIME (see later), HTTP servers can supply information of every possible data types;
- rapid: allows request-response cycles of less than 100 milliseconds

HTTP is not mandatory for distributed hypertext; there are other techniques and protocols that can be used to access or transfer information. However, like TCP/IP and HTML it is ubiquitous and so enables investment to develop e-commerce.

**Fields of Application**

The Web began as a tool to share knowledge and thoughts and have successfully evolved into a general communication machine between people and between systems. With the support of transactions and synchronous (time-ordered) communications, the Web has uses in many fields of applications.

A primary use is the dissemination of knowledge, which takes many forms. For example, chat rooms and bulletin boards are integral to interactive discussion of all kinds of subjects. Frequently Asked Questions (FAQs) published on Web sites offer answers to users’ questions on how to do certain kind of task. The variety of information that can be pulled out of the Web is very comprehensive.

Education includes a variation of the dissemination of knowledge. Open or distance learning programmes spearhead this aspect of the Web. Basically, any kind of demonstration on how to carry out certain tasks can be considered education. For example, a user can learn how to create a Web page from the numerous Web sites publishing such instructions.
With the possibilities of online trading, business transactions are carried out in the Web. The user simply keys in his orders and credit card numbers to buy products advertised on the Web. The Selling module would cover this subject area in depth.

The Web as Digital Library

The Web as a vast digital library is becoming what is known as a ‘Global Information Structure’. It will have a profound effect on how we live, work and play. We shall now look into a few of the social implications of the Web as a digital library and a marketplace.

Different Literacy

The hypermedia concept includes not only text and illustrations, but also music, animations, digital movies, video games and computer software. This diversity changes the form of literacy. The literacy to listen to music and watch a movie, is different from the literacy to read a book. Less literacy may be required with innovative ways of using this digital library. For example, software that reads text aloud can assist people with visual handicaps.

Indeterminate Quality and Value

Editors and publishers employing traditional methods of publishing, have little to gain from this type of publishing. As digital works can be copied at low costs, stored in almost no space and transported instantly anywhere in the world, writers can be their own publishers. Therefore, the works published are of indeterminate quality and value. Web publishing may provide no evaluation of work published.

Specialist Audiences

An article may perhaps interest a group of specialists in that field. With the Web, an average reader may browse through the article according to their degree of interest in the field. He or she may not want to be burdened with an additional flood of technicalities, or perhaps would navigate further to extract more in-depth information to supplement a deeper interest in the field.

Copyright Issues and Ease of Purchasing

The ease of copying digital works causes difficulties in protecting copyrights. It may be tempting to make illegal copies rather than finding the rightful owners and paying them a fee. On the other hand, the non-issue of distance, and the 24-hour, 365-day activity on the Web means that much can be easily bought through on-line shops. Consumers may come from distant areas of different time zones. With the Web, this marketplace is open at all times and can serve a very large global region. New technology even allows computational agents to staff this marketplace rather than people. Therefore, businesses are not constrained by distance and time variations.

Sense of Place

Despite the irrelevance of distance, an electronic marketplace may be attractive as it goes to the consumers instead of them physically moving to the business environment. Its sense of place is created as an illusion for the benefit of the consumers.

Benefits of Hypertext

We shall proceed with an analysis of hypertext documents.

Exercise 1

Write down your ideas about the possible benefits of hypertext using the following headings. If you like, go on-line to discuss these with colleagues before writing them down.
• Ease of insertion of new information
• Pointers to external materials
• Browsing

**Knowledge Additivity**

Links can be created to associate related subjects. Therefore the information given would be extensive and wide. The addition of two related subject areas is known as *knowledge additivity*.

Let’s say you want to find out how to tailor a shirt using a sewing machine. You would probably look in a book on tailoring a shirt and another on using a sewing machine. The information read would then be linked together in your brain. However, with the hypertext concept, this knowledge additivity would be simpler with association links. You can just continue clicking to read on both subject areas within the perceived single document.

**Drawbacks of Hypertext**

There are difficulties with working with hypertext, and so in providing or obtaining goods or services.

**Exercise 2**

Write down your ideas about the possible drawbacks of hypertext using the following headings. If you like, go on-line to discuss these with colleagues before writing them down.

• Navigation Difficulties
• No Main Catalogues
• Network Overload
• Link Fossilisation

Discussions and answers can be found at the end of the chapter

**Activity 1: Analyzing Hyperlinks**

Visit your favourite site and try to identify the following:

1. A chain of links
2. A loop
3. A guided tour

**Review**

Do Review Questions 1, 2, 3, 4

**The Client-server Computing Model**

When you are surfing the Web, you are using a web browser. When you go to a Web site for documents,
the site delivers them from called the Web server. The browser is considered to be a client in the relationship with the server as it is requesting information services from the Web. This is just one particular example of the client–server model of computing.

A Definition and some History

The client–server model has been defined as:

A software partitioning paradigm in which a distributed system is split between one or more server tasks which accept requests, according to some protocol, from (distributed) client tasks, asking for information or action. There may be either one centralized server or several distributed ones. This model allows clients and servers to be placed independently on nodes in a network.

Client–server computing is mainly about the client computer possessing its own computing power. In the days of mainframes, all the processing power took place on central computers. The client ‘terminals’ were little more than a television that could send and receive characters. When microprocessors became available, it was possible to make the terminals more powerful so that they could handle some of the processing. Over time this has meant that mainframes have been replaced by smaller server machines and terminals have been replaced by more powerful client workstations.

The client–server model provides a good division of processing power, since the server primarily provides information to the client which is responsible for interpreting and displaying it. This means that servers do not have to be powerful machines, allowing more people to become service providers.

A more important characteristic is that because client–server provides for significant processing power at the (remote) client end, the operator of the client system has considerable autonomous power in contributing to the enterprise of which he or she is a part. This means that local decisions can be made, and probably faster decisions made and actions taken.

You may hear client–server computing being talked about as a modern computing ‘paradigm’. Other than being part of a sales pitch, this is likely to mean that the model has made a significant impact on, and probably a significant change to, the way we design and use computer systems. In particular, it is the current model for distributed business systems, and so fitted nicely into the emerging Web.

Functionality

In the context of the Web, users run client programs (i.e. Web browsers) which provide the following functionality:

- They allow the user to send a request for information to the server.
- They format the request so that the server can understand it.
- They format the response from the server in a way that the user can read it.

Server programs carry out the following:

- They receive a request from a client and processes the request.
- They respond by sending the requested information back to the client.

Exercise 3

The client–server model applies to a lot of things outside of computers. Imagine going to a bank to withdraw some money? Who is the client and who is the server? Clearly, you are the client and the bank is
the server.

One of the advantages of the client server model is that one server can handle lots of clients. The teller in the bank (server) that you visit handles lots of other customers (clients). Also, you can use lots of different servers to get the service you need. (That is there are a lot of tellers, and for that matter, bank branches and cash machines.)

For any Web site, say the University of Cape Town Computer Science Web site [http://www.cs.uct.ac.za] or the Moodle site for the Department [http://moodle.cs.uct.ac.za] , think about the following questions and write down your answers:

1. Are there multiple clients?
2. Who are these clients?
3. Are there multiple servers?
4. Why would there be multiple servers?

Discussions and answers can be found at the end of the chapter

Information and Processing on the Web

Information is passed from the server to the browser. This information may be in the form of HTML documents, GIF files, Excel spreadsheets, movies – just about anything that is binary information.

Information can also be passed from the browser to the server. When you click on a hyperlink, you are sending information to the server, and when you fill in an online form, you are usually sending information to the server.

In addition to passing information backwards and forwards, some processing can also be done. For instance, you might have a simple Web page that calculates the overall cost of a loan once the initial value of the loan, the interest rate and the length of the loan have been entered.

But where does the processing take place? Does the server process the information and generate the result, or is it the client that processes the information? If the client does the processing, then this is a client-side application, and if it is the server, then it is a server-side application.

In the loan example above, the client has the information (the principle, rate and time). It could send this information to the server to process the information, generate the result and send it back to the client. Alternatively, the server could send the program to the client that will carry out the processing, and then the client could run it. In this latter case, since the client has all the information and the program is pretty small, it is probably better to run the application on the client side.

Of course, there is also a problem of who has the information. If the server has a database, and the client wants to query it, then there are two possibilities. The server could send the database and the querying program to the client to process it or the server could process it and simply send the result. In this case, it would probably be better to do the processing on the server side.

To summarize, where the processing is undertaken largely depends on where the information is, but also depends on the processing loads of the machines as well as the size of the program being run.

Exercise 4

On the East Med. Trading Co. Web site, they would like to display to the user the number of pages that he or she has visited at that site. Think about the following questions and make a note of your answers.
1. What data is needed?
2. Where is the data stored?
3. Should this be a client or a server side application

Discussions and answers can be found at the end of the chapter

**MIME Types**

A browser receives binary data from the server, which it has to cope with. How does it know if the binary data is an HTML document, a GIF picture file or something entirely different? Even if it does know what it is, how does it process it? The answer is MIME types.

MIME types –Multipurpose Internet Mail Extensions – were created to identify differing types of email attachment. These have been extended to include new multimedia types and are now used with a variety of protocols including HTTP. When information is sent to a browser, a MIME header identifies the file type of the document. Attaching a MIME type to a file allows the browser to process the file’s contents correctly without the browser having to guess at the data type from the file’s extension. Whilst MS-DOS files require a three letter extension to identify a file type, this is not true of all operating systems.

This MIME header information has the following format:

```
Content-type: type/subtype
```

where

- **type** is one of several general types such as: text, audio, image, video, application etc.
- **subtype** is a more specific designation. This is a large and ever expanding category.

Some examples of MIME headers are

- `text/HTML`
- `video/MPEG`
- `image/GIF`

**Processing MIME types**

Mime types are processed as follows.

1. Somehow the HTTP server must decide what type a file is. The server administrator can provide this information by instructing the server to map file extensions to certain file types. The server administrator must therefore supply a list of all the different file extensions of files found on the server and the equivalent MIME types for each of these files.

2. The client browser must also be configured to instruct it how to deal with these different types. Most browsers have been pre-configured, but you may need to amend them to cope with any new types you encounter. On Netscape Navigator 7.1, for instance, you use the Edit/Preferences… / Navigator/Helper Applications menu.

3. The client browser must also be configured to instruct it how to deal with these different types. Most browsers have been pre-configured, but you may need to amend them to cope with any new...
types you encounter. On Netscape Navigator 7.1, for instance, you use the Edit/Preferences… / Navigator/Helper Applications menu.

- View it in the browser. (Files such as GIFs, JPGs etc. can all be handled by the browser)
- Use a plug in. (Plug-ins are special pieces of code that software companies distribute to allow browsers to cope with new file formats.)
- Launch another application on your computer that can process the format.
- If all else fails, the file can be saved to disk until a suitable program is found.

**Exercise 1.5**

Imagine that there is a new document type that needs to be displayed on the client’s computer and so you need to introduce a new MIME type.

Use Microsoft Excel charts as an example (although they are obviously not new) and write down all the options to be considered on how this could be achieved.

Discussions and answers can be found at the end of the chapter

**Web Servers**

There are lots of different types of servers: FTP servers, gopher servers and, of course, Web servers. In this section we will concentrate on Web servers.

What do Web servers need to do? We will summarise their functionality here:

- They need to provide HTML pages (with the MIME type header).
- They need to provide other types of documents (with the MIME type header).
- They may need to process information from the user. For instance, if the user is submitting some information to the site, the Web server must either process and store that information, or pass it on to another program to do it.
- They supply dynamic data.

Processing user information and supplying dynamic data is complex. Many servers do not provide this facility. It makes for a much more complex system than one that simply provides data. But it also makes for a much more dynamic and useful server too.

User information can be processed on the server using server-side JavaScript applications, or CGI (Common Gateway Interface) scripts. Many other languages and interfaces are also used, e.g. Java Servelets and PHP.

The server will pass the user’s data to the JavaScript or CGI program, which will then process it. It may then dynamically create an HTML file, and send that file back to the client just as if it had already been stored. (Note, this will be discussed further in the units on JavaScript.)

**Distributed Processing**

Client–server computing is all about distributing the load of information and processing. Until about 20 years ago most information was stored on one computer and all the processing was done on that same
computer. The only reason an extra copy of the data might be kept, was for the likes of security or backup purposes. If a number of people needed the data or the processing, they would get another computer and copy the data.

With client–server computing, a given machine acts both as a client and as server, that is it can run both a Web server program and a browser client. It can also run processes (i.e. programs) on other machines. Network technology has enabled this distribution of processing and data.

The goal of distributing processing is to reduce the overall time that is needed to get some information. For example, one machine (named A) is connected to two other machines, B and C. If there are three processes to run, they can all run on A. If each takes 10 seconds, then it will take 30 seconds to run the processes. Now, if B and C are each asked to run a process, then the total processing time has now been distributed, and while it still takes 30 seconds of processing time, it only takes 10 real seconds of processing. It is therefore three times faster.

However, there is an additional cost that was overlooked in the above paragraph. If A has to ask B to run a process, there is some communication time. For instance, just sending a message takes some time, and that assumes that computer B has the right program and data to run the process. If not, A may have to send the data and possibly the program as well. Additionally, B has to send the answer back to A. (The same is true for Machine C.)

For simplicity sake, let us say that sending the data and the answer each takes 1 second. In second 1, A sends the data to B, A sends the data to C, and A starts processing. In second 2, B and C start processing. In second 10, A finishes processing. In second 11, B and C both finish processing and then send their responses to A. In second 12, A receives the data and everything is completed. The total time to run the three processes takes 12 seconds.

Now try some process balancing in the following exercise.

**Exercise 6**

The costs of data (including a program) sent over the net takes 1 second, and processing takes 10 seconds. All machines are directly connected to each other and are otherwise identical.

There are 5 computers. Computer A wants to run 4 programs (processes), computer B wants to run 2 programs, and computers C, D, and E are idle (no one is using them). The process times, location of the data and size of the data are described below. Machine A needs the answer from P1, P2, P3 and P4, and machine B needs the answer from P5 and P6. Transferring the answer should take 1 second.

<table>
<thead>
<tr>
<th>Program</th>
<th>Run Time</th>
<th>Location of Data</th>
<th>Size of Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>5 seconds</td>
<td>A</td>
<td>8 MB</td>
</tr>
<tr>
<td>P2</td>
<td>6 seconds</td>
<td>D and C</td>
<td>4 MB</td>
</tr>
<tr>
<td>P3</td>
<td>7 seconds</td>
<td>A and C</td>
<td>5 MB</td>
</tr>
<tr>
<td>P4</td>
<td>8 seconds</td>
<td>C</td>
<td>12 MB</td>
</tr>
<tr>
<td>P5</td>
<td>9 seconds</td>
<td>A and E</td>
<td>2 MB</td>
</tr>
<tr>
<td>P6</td>
<td>10 seconds</td>
<td>B</td>
<td>3 MB</td>
</tr>
</tbody>
</table>

1. Come up with 5 different ways of distributing the processing, and the total real time for processing. For example:
   - machine A runs P1 (5 seconds)
   - machine B runs P6 (10 seconds)
• machine C runs P3 (7 seconds plus one second to transfer the answer back to A for 8 seconds)
• machine D runs P4 (12 seconds to send the data from machine C, 8 seconds for processing, and
  1 second to send the answer for a total of 23 seconds)
• machine E runs P5 (9 seconds to run and 1 second to transfer the answer to B for a total of 10
  seconds)
• machine C runs P2 (6 seconds plus one for transfer for a total of 7)
• machine C is running both P2 and P3 so it takes 15 seconds.
• The longest time is for P4 which takes 23 seconds so the whole thing is done in 23 seconds.

2. What’s the smallest amount of time you can come up with to run all 6 processes and get the an-
swers back?

Hint. Think about the following questions:

• What does the data and processing represent?
• Can you give an example?
• Can you improve on 12 seconds?

Discussions and answers can be found at the end of the chapter

Review

Do Review Question 5 and 6

Review Questions

Review Question 1

Is it relatively simple to insert new information in hypertext?

Answer at the end of the chapter

Review Question 2

Is hypertext different from a hyperdocument?

Answer at the end of the chapter

Review Question 3

Explain the reason why it is difficult to retrieve required information with unlimited chaining of information.

Answer at the end of the chapter
Review Question 4

Explain why Knowledge additivity enhances the learning process.

Answer at the end of the chapter

Review Question 5

Here is a summary of what hypertext and hypermedia are about. Fill in the blanks.

A hypertext document is a __________ document that implements anchors containing ______ to con-
nect various fragments of information into one ______ network.

____________ is an extension to hypertext that includes digitised sounds and moving images.

The user is free to choose which links to follow in multi-linked hypertext document. Each sequence of
_______ constructs a unique __________ _____.

Authoring hypertext requires the decomposition of documents into fragments of information and then
the construction of links between the _____________.

The Web is an example of ________ hypermedia. There is no central authority _________ its devel-
opment and its information content is geographically-___________. The ease of linking information is
one of the major benefits of hypertext.

Navigation difficulties through the various possibilities of different _____ is the main drawback of hy-
pertext. Extracting the information required can become tedious.

Discussions and answers can be found at the end of the chapter

Review Question 6

What is so important about the client–server model of computing?

Discussions and answers can be found at the end of the chapter

Answers

Discussions for Exercise 1.2

The following are some thoughts on usefulness of hypertext

Navigation Difficulties

Navigation is the main drawback of hypertext. As the document is interlinked and may loop, readers can
easily lose track of where they have been and where they are. The freedom to choose any of the links to
follow on may take the reader away from the item being sought. There is insufficient clue to where an
anchor would link.

No Main Catalogues

A catalogue is readily available in a physical library and the user can easily find out whether the book he
requires is available or not. It is difficult to index a hypertext document due to multiple links within a
document, unless the reader is guided to a certain sequence of links.
Therefore, extraction of information required is more tedious especially in large hypertext documents. Search facilities attempt to reduce this practical difficulty. However, when these deal with a large content of information such as the Web, the result of the search itself may prove to be rather extensive too. Moreover, these search facilities recognise only text description and not the multimedia content of the document. This undermines hypermedia.

**Link Fossilisation**

The name of the computers may change and linked documents may be moved to other host computers. Hypertext requires explicit pointers to the names of the computers and the files on these computers. The fragment of information that is linked will no longer be accessible, or perhaps become outdated as the information is of the old version and the link not updated with the new version of information.

**Network Overload**

Hypertext content assumes universal coverage and infinite transfer capacity. The capacity of the telecommunications network may not be sufficient to cope with the usage, without penalising or compromising other network activities. This happens when the reader is not warned of the document size or is not conscious of the network implications. What results then is technical halt or slowdown to navigating through the hypertext.

**Answers to Exercise 1.3**

1. Multiple Clients
2. These are all the people who visit it, or more precisely, they are the browsers those people are using.
3. There are not multiple servers for this site.
4. For heavily used sites, the site is copied to another address. This is called a mirror site. It is used to reduce traffic to the base site, and overall net traffic, as clients will go to a mirror that is closer than the base server. This also adds redundancy and makes the site more likely to be up. If one site goes down, one of the mirrors is still likely to remain up.

**Answers to Exercise 1.4**

The following might be needed to give a history of site usage.

1. The data that is needed is a list of all the pages that a given browser has visited on the site.
2. What the server can do is maintain a list of all the people who visit the site and which pages they have visited. This can be maintained (i.e. stored) on the server.
3. Alternatively, some JavaScript code could be used to maintain that information on the browser using cookies. This is probably more difficult, and would only work for the current visit. (When the browser application was halted, all the information would be lost.) So, it would be better to store the information on the server side.
4. If the data is stored on the server, then this should be a server-side application. The server can process the information and simply return a number representing the number of pages this user has visited. That number is returned to the browser. The alternative is to send all of the data (all the pages/user pairs) to the browser and then carry out the processing there.
Answers to Exercise 1.5

The server will append the new type onto the information it sends. The server needs to know the type, but that should be fairly straightforward as the server is providing the data, so the administrator will have set that up.

The client needs to know what the type is, and have a method of displaying the data.

The browser itself could display the data, e.g. like a GIF file.

The browser could use a plug-in. If it were a new file type, the group who developed the file type might provide a plug-in to read the type.

The other options are automatically invoked from current browsers. Since the browser does not know what to do with the file type, it puts up a dialog box to ask the user. If the user knows which application to invoke, the user names that application; which is then invoked and passed the file (the Excel chart file in this case). Instead of opening the file, you could execute Excel, open the file, save it, and deal with it later.

Answers to Exercise 1.6

Discuss the results of Activity 5 with your colleagues studying the module in the Discussion Forum. In particular, you should:

• state the smallest amount of time you were able to run all 6 processes and be able to explain your results
• discuss what the data and processing represents
• and be able to give an example

Answers to Review Question 1.1

It is easy to create anchors which will link fragments of information together in a hypertext. There is no real sequence to the information. New fragments can be inserted anywhere in a hypertext document as long as the anchors are properly implemented to link the new to existing fragments.

Answers to Review Question 1.2

Hyperdocuments are hypertext documents. They are the same thing.

Answers to Review Question 1.3

If the hypertext document is small and does not contain many external links, information can be retrieved quickly with the browsing feature. The difficulty arises when the hyperspace is vast and there are many links in each page. Numerous links imply the possibilities of many different exploration paths. This makes navigation through the network of documents for the required information tedious.

Answers to Review Question 1.4

Knowledge additivity connects different aspects of information from different fields of study together, therefore the information is more useful. Let’s say X=hunting skills and Y=using bows and arrow. Take a reader who wants to know how to hunt well with bows and arrows (Z). It is possible to achieve this with knowledge additivity in hypertext. (Z=X+Y)
Answers to Review Question 1.5

A hypertext document is a non-linear document that implements anchors containing links to connect various fragments of information into one mesh network.

Hypermedia is an extension to hypertext that includes digitised sounds and moving images.

The user is free to choose which links to follow in multi-linked hypertext document. Each sequence of links constructs a unique navigation path.

Authoring hypertext requires the decomposition of documents into fragments of information and then the construction of links between the fragments.

The Web is an example of networked hypermedia. There is no central authority dictating its development and its information content is geographically-independent. The ease of linking information is one of the major benefits of hypertext.

Navigation difficulties through the various possibilities of different paths is the main drawback of hypertext. Extracting the information required can become tedious.

Answers to Review Question 1.6

It is important because it allows for significant processing power at the remote client end so that the operator of the client system has considerable autonomous power in contributing to the enterprise of which he or she is a part.