CHAPTER 1

UNDERSTANDING HYPERTEXT SYSTEMS

CHAPTER OBJECTIVES

In this chapter, you will learn about:

Technical descriptions of the Web are filled with references to *hypertext*. *HyperText* Markup Language (HTML), *HyperText* Transport Protocol (HTTP), and *hypertext* reference (href) are but a few. Even in a formal definition, the WWW is categorized as "a networked *hypertext* system."

Before we say what hypertext is, it is perhaps more important to say what it isn't. A popular quote in the hypertext community is, "Hypertext is *not* simply linear text with links, any more than television is simply radio with a picture." Likewise, many authors' interpretation of hypertext is that "hypertext is more than point and click."

Hypertext is an authoring and reading paradigm that long predates the Web and for which a substantial body of research exists. Thanks to the success of WWW, hypertext has also become a new medium of expression and should be used accordingly. A new medium not only dictates who the audience is, but also defines what the content should be and how it should be presented.

In this chapter you will learn the fundamental concepts of hypertext and hypertext systems. The goal is to illustrate the unique communication medium that hypertext represents and how it can best be used by content authors/providers and users/readers. We will also discuss how some basic hypertext techniques can be used in the development of Web-based content and how these techniques address some of the fundamental Web-authoring issues. All along we will be evaluating the effectiveness of WWW as a hypertext system with emphasis on what the technology provides and what an author must provide.

LAB 1.1

DOCUMENT MARKUP FUNDAMENTALS

LAB OBJECTIVES

After this lab, you will be able to:

- ✔ Explore the Concept of Document Markup
- ✔ Apply All Five Types of Document Markup to HTML
- ✔ Define a Document Tagset
- ✔ Explore the Issues Surrounding HTML Page Conversion

A discussion of document markup is relevant to Web content development because markup is a critical component of the authoring process. Most Web authors are familiar with document markup using HTML (the HyperText Markup Language) and maybe XML (eXtensible Markup Language); however, few are familiar with the fundamentals of document markup and the types of markup that have been defined.

By definition, document markup is not a part of the intellectual content of a written work, but instead provides information about how that work is structured and how it should be interpreted, read, presented, or displayed. Although independent of a work's content, markup can change a reader's interpretation of that content.

In the planning phase of document development, an author must make a conscious decision to separate content and structure from presentation.

- Content is "what" you say.
- Structure is how you organize what you say.
- Presentation is how what you say and how you organize the information appears, feels, or sounds to your audience.

Five types of document markup are used to help make these distinctions:

1. Punctuational
2. Presentational
3. Procedural
4. Descriptive
5. Referential

Each of these types will be discussed and examples will be given.

TYPES OF DOCUMENT MARKUP

PUNCTUATIONAL MARKUP

Punctuational markup consists of the use of a well-defined set of punctuation marks to provide syntactic or rhetorical information about how some body of text will read. Punctuational markup requires some level of agreement on how the markup is appropriately used.

■ *FOR EXAMPLE*

There are precise literary rules surrounding the use of the comma. Probably very few of us know these rules and use the comma when "it feels right," or when a pause should occur during reading. It is clear that the omission or incorrect use of a comma can change the interpretation of a sentence dramatically.

PRESENTATIONAL MARKUP

In presentational markup, an author introduces elements intended to make the organization of the work clearer to a reader. These elements are often visual or structural in nature.

■ *FOR EXAMPLE*

Examples of presentational markup are page layout definitions (horizontal and vertical space and margins) and page breaks to occur during printing. Markup that supports structural definitions such as chapter, section, subsection, and so on could also be viewed as presentational.

PROCEDURAL MARKUP

A procedural markup system consists of a collection of commands that allow an author to specify how text and other document components are to be formatted. Procedural markup can, and often does, replace presentational markup.

■ *FOR EXAMPLE*

Good examples of procedural markup are the TEX and LATEX systems. For example, in LATEX, a fraction in a mathematical expression is "marked up" by an author as \frac{numer}{denom} where numer and denom are the values of the fraction's numerator and denominator respectively.

DESCRIPTIVE MARKUP

In descriptive markup, document authors identify the element type of text tokens or collections of text. The scope of these types is then tagged in order to insure that the affected text is presented in the desired manner.

■ *FOR EXAMPLE*

It should be fairly obvious that HTML and XML are good examples of a descriptive markup system. As is familiar to all HTML and XML authors, *tagsets* (e.g., <H1> and </H1>, <OL> and </OL>) form *containers* that indicate how the enclosed text and/or objects are to be presented/rendered by a Web browser. Some tags have no scope (e.g.,
) because they indicate an immediate action and do not contain text or objects.

REFERENTIAL MARKUP

Referential markup allows an author to insert external entities into a document. This is accomplished by including references within the document that result in the replacement with the entity when the document is processed.

■ *FOR EXAMPLE*

Once again, HTML contains examples of referential markup. Since browsers are designed to "collapse" strings of "white space" (e.g., spaces or tabs), an author who deliberately wants to include three spaces in some text cannot just type in three spaces. It would be necessary for that author to enter the entity   (nonbreaking space) three times. Another entity example would be á if the author needs an acute accented "a." Therefore, the   reference results in the inclusion of three spaces when the text is rendered and the á entity results in inclusion of the appropriately accented "a."

The HTML <IMG> tag is referential markup in that it specifies the necessary parameters for the inclusion of an external entity (i.e., an image) into a page.

LAB 1.1

EXAMPLES OF MARKUP LANGUAGES USED IN WWW

In our discussion of the types of markup languages, it can be seen that each of the described types addresses particular author needs and/or concerns. We've also seen that a markup language such as HTML does, in fact, include multiple markup features. A Web author would want to have access to as many markup features as possible in order to create the richest and most visually "correct" document. It is well known that HTML is a weak markup tool—its major advantage is that it is easy to learn. It is also ubiquitous and highly portable. However, HTML is unable to adequately express or contain structures (e.g., chapters), nor is it easily tailorable for different users, contexts, or media. Considerable effort is underway to enhance the descriptive "power" of HTML.

In this section we will discuss three markup systems that also have particular relevance to Web authoring:

- SGML—Standard Generalized Markup Language
- XML—Extensible Markup Language (or eXtensible Markup Language)
- MathML—Mathematics Markup Language

Chapter 7, Overview of Multimedia Formats, includes a brief description of VRML, one of the technologies for bringing virtual reality to the Web. VRML does *not* stand for Virtual Reality Markup Language, but rather Virtual Reality *Modeling* Language. VRML is used to describe and model virtual environments through which Web users can navigate.

SGML

In terms of Web/Internet years, SGML has been around for a long time. It is an international standard (ISO 8879) that defines, in general, an interchange format for electronic documents. One of the key features of SGML is reusability. This means that it allows the definition of a document that is platform (computer) and operating system independent. It also means that the version of a document used for reading online can also be used by a publisher for a printed version.

SGML is actually a meta-language (a language used to define languages) capable of defining an infinite (theoretically) number of document types. That means that you can use SGML to describe layout unique to a brochure or a newspaper/newsletter or a billboard. A particular document type is defined in a Document Type Definition (DTD) written in SGML. The DTD defines the permissible markup tagsets.

SGML was in widespread use for document description at CERN (the European Laboratory for Particle Physics) when Tim Berners-Lee invented the World Wide

Web (WWW) there. He realized that he would need a markup language capable of defining the elements unique to hypertext pages. So, he wrote an SGML DTD that included those elements. That SGML DTD is HTML. HTML and other DTDs are known as SGML *instances*.

A document of an SGML-based type references the DTD defining its tagset. For example, the <!DOCTYPE> tag that HTML authors *should* include at the top of their page files (but usually don't) is actually an SGML tag specifying the version of the HTML DTD to be used in processing the file.

XML

As the Web has gained more widespread use, authors have demanded greater control over document layout and structure. It has become increasingly clear that HTML is unable to provide the precise control desired. So, why not put SGML on the Web, since it offers infinite control?

Given the extent of its capabilities, it should be no surprise that SGML is quite complex. Writing an SGML DTD is a sophisticated task. Bringing SGML to the Web would conflict with the "ease of use" of HTML that has allowed so many authors to publish on the Web.

In 1998, a committee of the World Wide Web Consortium (usually called the W3C) released a definition for XML (Extensible Markup Language or sometimes eXtensible Markup Language). XML has been described as "SGML-lite" because it has stripped away the "heavy elements" of SGML and concentrated on those features most relevant to Web authoring.

Like SGML, XML *schemas* describe specific document types. But unlike SGML, these instances/DTDs are easier to define. XML also addresses other Web-specific issues such as providing an expanded capability for program use of XML files (e.g., search applications that can retrieve and exchange information). It also supports internationalization efforts that will allow Web pages to be displayed not only in languages other than English, but in languages using other than the Latin-Roman alphabet (e.g., Chinese, Arabic, Hebrew).

While there are numerous books and articles about XML, future study should begin at the W3C website—http://www.w3.org/XML/.

MathML (MML)

Since the Web was developed in a scientific environment, there has been a desire since the beginning for support of mathematical formatting. Early efforts were unsuccessful. Scientific authors have used various (and nonstandard) methods

for including mathematical formulae in their pages. Mathematical expressions are often included by means of graphical images produced by other applications such as TEX or Mathematica. Some authors have used third-party multimedia applications that require browser plug-ins. While such methods have enjoyed limited success, pages using them are usually limited to readers with graphical (i.e., non-linemode or text) browsers or with access to the required plug-in.

MathML is an XML-defined schema for describing mathematical expressions. The same description will display appropriately on both graphical and linemode browsers. In addition, applications are being developed that will allow readers to "cut and paste" MML expressions directly into mathematical software for calculation and/or graphing.

MathML efforts are in a continuous state of flux. In the meantime, several other technologies have been introduced to support mathematical formatting.

As with most Web technologies, the first stop for future study (and progress reports) on MathML should be the W3C website—http://www.w3.org/Math/.

HTML CONVERSION

The "rush" to get on the Web has led to the development of numerous tools to support the conversion of existing documents to HTML. Examples of such tools are:

- LaTEX2HTML—for converting LaTEX documents to HTML
- rtf2HTML—for converting Rich Text Format (RTF) files to HTML
- ps2HTML—for converting PostScript files to HTML

We have also seen word-processing and text-processing applications (e.g., Microsoft Word) begin to offer a "Save as HTML" option. Word processor formats as well as other markup formats such as LaTEX offer a much greater range of formatting options as compared to HTML. In addition, you have much finer control over the formatting that HTML provides.

LAB 1.1 EXERCISES

1.1.1 EXPLORE THE CONCEPT OF DOCUMENT MARKUP

a) Compose a sentence where punctuational markup (e.g., the use of commas) actually changes the meaning or interpretation.

b) What appearance aspects of a book would be considered the result of descriptive markup?

1.1.2 APPLY ALL FIVE TYPES OF DOCUMENT MARKUP TO HTML

We have described HTML as containing components corresponding to all five types of document markup—punctuational, presentational, procedural, descriptive, and referential.

a) Give examples of HTML for each of the five types of markup.

1.1.3 DEFINE A DOCUMENT TAGSET

We have indicated that part of the power of SGML and XML is the ability to describe a markup tagset unique to a specific type of document. Suppose you are an entrepreneur and you want to define a generic description for an online billing invoice.

a) What are the common elements of an invoice that could be defined by a presentational tagset (e.g., <amountdue>)?

1.1.4 EXPLORE THE ISSUES SURROUNDING HTML PAGE CONVERSION

The availability of tools to convert other formats can present dilemmas for the Web author. Authors constantly ask themselves if they should convert an existing document to HTML or reauthor that document in HTML. Take a word processor that can store in both RTF and HTML (for example, recent versions of MS-Word or Corel WordPerfect). Create a fairly complex document with various fonts and format elements (e.g., tables, headers, and so forth).

a) Load the HTML page into a browser and compare the appearance to what is in the word processor. What differences do you see?

b) Load both the HTML page and the RTF page into a text editor such as Notepad. Try to figure out where the text elements are that you included as well as the markup components. What can you say about each?

c) Comparing the contents, what issues arise when deciding whether to author directly in HTML or convert text from another format?

LAB 1.1 EXERCISE ANSWERS

1.1.1 ANSWERS

a) Compose a sentence where punctuational markup (e.g., the use of commas) actually changes the meaning or interpretation.

Answer: There can obviously be an unlimited number of answers to this question.

For example, these sentences are identical in content but have different meanings due to their punctuational markup:

"Tom," says Tim, "is wrong."

Tom says, "Tim is wrong."

Although less obvious, HTML provides punctuation markup to a limited extent. For example, the <QUOTE></QUOTE> tag set would be used to change the appearance of the text indicating it would be a quote from someone. Without this markup, the text may be considered part of the document and not coming from a different source, for example. Note that in the context of HTML, this is also procedural markup because it tells the browser to change its behavior when displaying the text.

b) What appearance aspects of a book would be considered the result of descriptive markup?

Answer: In the lab, we discussed the <H1></H1> tag set as being a "container," because it contains certain text. In this case, it is also describing that text as a level 1 header. Books also have similar contents and therefore similar markup. Other aspects of a descriptive mark would be the subheading contents, illustration captions, and so forth. If you change the appearance associated with the description "Chapter Heading," all of the chapter headings change.

As an interesting side note, Corel WordPerfect enables you to see and manipulate the markup directly. Therefore, you can much more easily see why certain text is being displayed they way it is (and not the way you want it).

1.1.2 ANSWER

a) Give examples of HTML for each of the five types of markup.

Answer: There is no one specific answer to this question, but the following offer some examples. Compare your answers to the types of HTML tags given.

- ***Punctuational:*** *HTML does not contain any tagsets that could uniquely be described as punctuational markup. HTML authors are free to include punctuation (e.g., periods, commas, etc.) wherever they choose.*
- ***Presentational:*** *The HTML tag <HR> is presentational in that it specifies the inclusion of a construct (a horizontal rule) whose sole purpose is to separate document/page content.*
- ***Procedural:*** *The HTML <FORM> and <TABLE> tagsets are procedural in that they support unique page/document components. Certain collections of tags (e.g., <INPUT>, <TD>) can only be used within these components.*
- ***Descriptive:*** *Most of the HTML tagsets are descriptive. For example, the header tags (<H1>, <H2>, etc.) specify how specific text is described with respect to other text in the document/page.*
- ***Referential:*** *The HTML 4.0 <SCRIPT>, <OBJECT>, <STYLE>, and <LINK> tags can "refer" to external entities that help to form a total description of the page/document that contains them.*

1.1.3 ANSWER

a) What are the common elements of an invoice that could be defined by a presentational tagset (e.g., <amountdue>)?

Answer: The following are some of the data entities you would expect to find on a billing invoice:

```
<invoicedate>
<orderdate>
<invoicenumber>
<customerlastname>
<customerfirstname>
<customertitle>
<customerorganization>
<organizationaddress>
<amountdue>
<productorservice>
<duedate>
```

Note here that since these are parts of a tagset, there is a closing tag indicating the end of the element. For example, there would be an </invoicedate> and a </duedate>, as well.

Although one could create tags and tagsets that are short, and therefore easier to type, the ones listed here intuitively tell us what elements they describe. We could then create

a template using these tags and have an external program (such as a Web browser) load the template and then pull the real values from a database, for example.

The DTD that described this particular document could go so far as to say that particular tags are required (like <invoicenumber>), whereas others are optional, like (<customertitle>). Take a look as the tags available for HTML and see which ones are required and when.

1.1.4 ANSWERS

a) Load the HTML page into a browser and compare the appearance to what is in the word processor. What differences do you see?

Answer: Depending on your document, the original document will have a much more varied appearance, which will probably be closer to the way you had envisioned the original document. The HTML conversion of the word processor may not be able to assign HTML tags to all of the elements you chose or in the way you chose them and may either make "guesses" or leave things out.

b) Load both the HTML page and the RTF into a text editor such as Notepad. Try to figure out where the text elements are that you included as well as the markup components. What can you say about each?

Answer: Depending on your document, there will be a lot of information in the RTF file in the header (at the top of the file), which may cause "information overload." If you look carefully you will find your text elements in there. With the HTML file, the textual information is much easier to find, and interpreting what each of the tags does is generally much easier. Looking at these two documents, it should be clear why people do not typically edit RTF files directly, but rather use a word processor.

c) Comparing the contents, what issues arise when deciding whether to author directly in HTML or convert text from another format?

Answer: Conversion tools usually involve "stepping down" from a "richer" document description format (e.g., RTF) to a "poorer" description format (HTML). No matter how good the tool is, compromises must be made. These compromises usually affect presentation at some level. Some tools have been known to use tricks (e.g., using HTML tables for layout) or browser-dependent tagsets. The resulting HTML files may therefore have limited usability.

These tools may be good for small documents, in which case the author can still "tweak" the HTML markup if necessary. Editing HTML directly does not appeal to everyone, but even programs that are specifically designed to create HTML documents often do not allow constructs that can be created by editing the documents directly as well as displayed in every browser. Some examples I have encountered are tables within forms or bulleted headers.

LAB 1.1

LAB 1.1 SELF-REVIEW QUESTIONS

In order to test your progress, you should be able to answer the following questions.

1) HTML and XML are instances of SGML.

a) _____ True
b) _____ False

2) Procedural Markup always implies a programming language like JavaScript or Active Server Pages.

a) _____ True
b) _____ False

3) HyperText is so named because it transmits the data across the "hyper media" of the internet.

a) _____ True
b) _____ False

4) DTD stands for which of the following?

a) _____ Data Transmission Document
b) _____ Document Transmission Definition
c) _____ Document Type Definition
d) _____ Document Transmission Data

5) Which of these is not a reason for using a "rich" text document (such as MS-Word) as compared to HTML?

a) _____ Finer control of the appearance
b) _____ Wide range of formatting elements
c) _____ Ability to have "forms"
d) _____ Document easily read and changed by hand

6) Tags that describe the appearance of the document are known as which of the following?

a) _____ Punctuational
b) _____ Presentational
c) _____ Procedural
d) _____ Descriptive
e) _____ Referential

7) Tags that refer to a characteristic of a particular elements are known as which of the following?

a) _____ Punctuational
b) _____ Presentational
c) _____ Procedural
d) _____ Descriptive
e) _____ Referential

Quiz answers appear in the Appendix, Section 1.1.

LAB 1.2

HYPERTEXT CONCEPTS

LAB 1.2

Spoken language is a series of words, and so is conventional writing. We are used to sequential writing and so we come easily to suppose that writing is intrinsically sequential. It need not be and should not be.

Many people believe these forms of writing [hypertext] to be new and drastic and threatening. However, I would like to take the position that hypertext is fundamentally traditional and in the mainstream of literature. Customary writing chooses one expository sequence from among the possible myriad; hypertext allows many, all available to the reader.

—Ted Nelson, 1993

LAB OBJECTIVES

After this lab, you will be able to:

- ✔ Explore the Nonlinear Nature of Hypertext
- ✔ Identify Examples of Nonlinear Text
- ✔ Define Hypertext and Hypermedia

In 1965, Ted Nelson coined the term *hypertext* to describe "non-sequential or non-linear text." He defined it as "a body of written or pictorial material interconnected in a complex way that could not be conveniently represented on paper." Nelson (and others before him) was attempting to define an authoring and reading style that more closely approximates the way people think. In particular, written text (such as books or newspapers) are linear in nature. They are written sequentially and meant to be read sequentially. On the other hand, we do not think sequentially—our thoughts often jump between multiple ideas,

concepts, and references that our brain manages to bring together into a meaningful whole. Reading and writing processes are based on the nonlinear nature of thinking.

It is difficult to accurately represent the way we think using written text due to the limitations of the medium (paper). However, specific literary styles and conventions have evolved that attempt to address (in part) the nonsequential issue. For example, the table of contents and index of a book allow the reader to "jump" into the middle of the text in order to locate specific information. They allow books to be used nonsequentially (imagine having to find something in a book by searching all pages in order!). Footnotes allow an author to provide additional information "out of the usual document flow." To read a footnote, the reader interrupts a linear reading process. Reading a footnote is usually optional.

Other authoring styles that support nonlinear reading are more dramatic and not as well standardized as tables of contents or indices. Textbooks often have multiple sections that can be read or skipped based on the reader's level of expertise or interest. A popular children's series, "Choose Your Own Adventure," allows readers to choose alternative paths through a book, which may result in different story endings. Both of these examples place emphasis on the reading audience's thinking and learning styles.

Those of you who are serious students of computer-based hypertext should be sure to read books by Ted (Theodor Holm) Nelson—The Literary Machine and Computer Lib/Dream Machines *(1987) and* Literary Machines 93.1 *(1993). An excellent scholarly work on hypertext is* Hypertext: The Convergence of Critical Theory and Technology *by George Landow. This book is also available on CD so that it can be read in a "hypertextual" fashion.*

Another great source for information on hypertext is the ACM Special Interest Group (SIG) SIGWEB (http://www.acm.org/siglink/).

LAB 1.2 EXERCISES

1.2.1 EXPLORE THE NONLINEAR NATURE OF HYPERTEXT

We have described a book index as being a "weak" form of hypertext because it allows a user/reader to "jump" into a book nonsequentially. However, indices are by their very nature set up alphabetically. People don't think alphabetically and alphabets are a characteristic of the language being used.

a) Could there be a better way of providing index-type information more similar to the way we think?

1.2.2 IDENTIFY EXAMPLES OF NONLINEAR TEXT

Understanding how nonlinear text has been used can help to understand how it might be used. Ted Nelson cites numerous historical and religious examples of nonlinear literature (hypertext).

a) Either research Nelson's references or identify some examples of nonlinear literature yourself.

1.2.3 DEFINE HYPERTEXT AND HYPERMEDIA

Hypermedia brings the nonlinear associations to multimedia that hypertext brings to text. Hypertext that includes nontextual content (such as images or audio) is actually hypermedia.

a) Expand the definition of hypertext to include hypermedia and describe an example of hypermedia.

LAB 1.2 EXERCISE ANSWERS

1.2.1 ANSWER

a) Could there be a better way of providing index-type information more similar to the way we think?

Answer: A "hyper-index" might allow a user to identify relevant document content by supporting an interactive search of keywords, concepts, descriptions, and so forth. The user would then not be constrained by the alphabetical ordering of the index or by the arbitrary selection of index entries.

One existing text-based example would be the Encyclopedia Britannica, with its "Outline of Knowledge." Although information is listed alphabetically, it is broken down by concepts and not specific entries. Many technical documents online have hyperlinks that lead to other areas of the text, such as a detailed discussion of something simple being mentioned, glossaries, or additional information.

1.2.2 ANSWER

a) Either research Nelson's references or identify some examples of nonlinear literature yourself.

Answer: Nelson suggests that nonsequential writing has roots that extend into antiquity. The Talmud, with its use of annotations and nested commentary, is a classical example. The Indian epics such as Ramayana and Mahabharata are comprised of stories branching off to other stories.

William Dickey, a poet who works with hypertext, suggests that authors create links that offer several sets of distinct reading paths: "The poem may be designed in a pattern of nested squares, as a group of chained circles, as a braid of different visual and graphic themes, as a double helix. The poem may present a single main sequence from which word or image associations lead into sub-sequences and then return."

There is a whole study of hypertext literature, as well as nonlinear literature, on the Hyperizons Web page (http://www.duke.edu/~mshumate/hyperfic.html). This also includes things like collaborative literature. There are also a number of links at the Audiovisual Institute of Pompeu Fabra University in Barcelona, Spain (starting at http://www.iua.upf.es/literatura-interactiva/eng/p5.htm). There are lots of interesting examples if you're inclined to research this. One particular example, found at http://192.211.16.13/curricular/panopticon/student_projects/fiction/thread.htm, presents an interesting implementation of original collaborative hypertext fiction called CityThreads. It tells the story

of a number of characters in the Seattle area on the same day. There is no beginning or ending—each story details the life of one of the characters and contains links to the other characters' stories whenever two characters meet. While each story may be read linearly, the reader will achieve the most enjoyment by exploring the story through the links.

1.2.3 ANSWER

a) Expand the definition of hypertext to include hypermedia and describe an example of hypermedia.

Answer: Hypermedia is an extension of the idea of hypertext that includes multimedia. Since text is one of the media types supported in a multimedia system, then hypermedia incorporates hypertext.

In a hypermedia document, the author creates links (or associations) between multimedia components. An example of hypermedia could be a textual description that links to an audiovisual presentation.

LAB 1.2 SELF-REVIEW QUESTIONS

In order to test your progress, you should be able to answer the following questions.

1) Hypertext is generally considered "nonlinear" because you can move the document in many different directions.

a) _____ True
b) _____ False

2) A footnote provides linear information because it appears on the same page as its reference.

a) _____ True
b) _____ False

3) Indexes provide hypertext-like features to written media because you can "jump" directly to specific text.

a) _____ True
b) _____ False

4) Which of the following would be nonlinear methods of accessing information?

a) _____ index
b) _____ Web Search engine like Yahoo
c) _____ table of contents
d) _____ all of the above

5) Which of the following would be not be a method to provide nonlinear content to a document?

a) _____ hyperlinks
b) _____ in-line images
c) _____ scripting languages like JavaScript
d) _____ search pages

Quiz answers appear in the Appendix, Section 1.2.

LAB 1.3

THE FUNDAMENTAL COMPONENTS OF A HYPERTEXT SYSTEM

LAB OBJECTIVES

After this lab, you will be able to:

- ✔ Identify the Difference Between the Concepts of Hypertext and a Hypertext System
- ✔ Explore Computer-Based Hypertext Systems
- ✔ Explore Web-Based Hypertext Systems

If human thought is basically nonlinear, then we can define the human learning and perceptual process as essentially organized as a semantic network in which concepts are linked together by associations. We learn and remember through nonlinear associations. Likewise, the fundamental components of a hypertext system are:

- Nodes—representing concepts
- Links—representing the relationships between nodes

Graphically, a particular (relating to a particular body of information) collection of nodes and links (a hypertext system) might be represented as shown in Figure1.1.

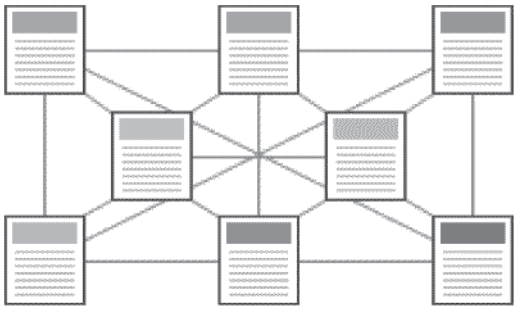

Figure 1.1 ■ An Example of a Hypertext System

In Figure 1.1, the pages represent the *nodes* and the lines represent the *links*. It is not clear from this graph the path that a reader would follow when "reading" this system.

NODES

A node is defined as a single concept or idea. As such, nodes can be virtually any kind of information. In human thought, a node might be a sound or a smell or an experience. In a computer-based hypertext system, a node can contain text, graphics, animation, audio, video, images, programs, and so on. Nodes can be "typed," indicating how they are used. For example, if a node in a Web-based hypertext system is designated as the "home page," there is the implication that that node will be used in a specific way to traverse that system (i.e., it is the node where readers will begin).

Finally, nodes are connected to other nodes with links. The node from which a link originates is called the *reference;* the node at which a link ends is called the *referent*. References and referents are also called *anchors*. The contents of a node are made available (viewed, remembered) by activating a link.

LINKS

The role of a link in a hypertext system is to connect related concepts or nodes. In general (though not in WWW), links are bidirectional, meaning that a reader can go backwards and forwards. The word *beach* could be a link invoking a vision of warm sand (a node/concept) and the sound of the surf (another node/concept). "Activating" a link "reveals" the content of a node.

Like nodes, links can also be "typed," illustrating features of the relationship of the nodes they connect. For example, a link might be simply a reference (like the word "beach" above) or actually reflect some relationship between nodes such as parent–child, chapter 1–chapter 2, and so on.

COMPUTER-BASED HYPERTEXT SYSTEMS

In general, computer-based hypertext systems contain the following elements: a user interface (UI), an authoring system, an information retrieval (IR) system, a hypermedia system, and a storage system. These systems should be used in determining the overall hypertext plan or roadmap to navigating through your Web site:

- A user-interface that assists the reader/user in moving (navigating) through large amounts of information; the user-interface provides the reader with the ability to activate links and read the contents of nodes/pages. This is often a Graphical User Interface (GUI).
- An authoring system providing tools to create and manage nodes (of multiple media) and links.
- Traditional information retrieval (IR) mechanisms such as keyword searches, author searches, and so on. There are also attempts to incorporate structure queries along with content queries—retrieving a part of the hypertext network based on some user-specified criteria.
- A hypermedia engine used to manage information (e.g., a database) about nodes and links.
- A storage system that can be a file system, a knowledge base, a relational database management system, or an object-oriented database management system.

LAB 1.3 EXERCISES

1.3.1 IDENTIFY THE DIFFERENCE BETWEEN THE CONCEPTS OF HYPERTEXT AND A HYPERTEXT SYSTEM

Lab 1.2 described hypertext concepts. This lab described hypertext systems.

a) Must "hypertext" always be delivered by a "hypertext system"?

In the "Choose Your Own Adventure" series, the book text is the "hypertext."

b) What is the "hypertext system" of the "Choose Your Own Adventure" series?

1.3.2 Explore Computer-Based Hypertext Systems

Implementation of a "hypertext system" using computers and networks is well understood. However, one of the first (and best-known) hypertext systems—Memex—was based on microfilm. Research the work of Vannevar Bush and the Memex system.

a) What was the Memex system intended to be?

b) List and discuss existing computer-based hypertext systems.

1.3.3 Explore Web-Based Hypertext Systems

WWW is certainly the world's most famous and successful hypertext system. But how good is Web hypertext?

a) Compare the Web's "hypertextual" features (e.g., node and link definition, navigation, and so forth) with some of its lesser-known predecessors and peers such as Xanadu, Zog, or Hyper-G.

LAB 1.3 EXERCISE ANSWERS

1.3.1 ANSWERS

a) Must "hypertext" always be delivered by a "hypertext system"?

Answer: When one thinks of a "hypertext system," normally there is the belief that some "device" delivers the hypertext to the user. However, the "Choose Your Own Adventure" series is one example where there is no "device" in the traditional sense. Simply, choices made by the reader determine the "path" the story takes. Looking at it abstractly, something like the Encyclopedia Britannica is a hypertext system as there is no requirement to traverse the information linearly and the reader is free to choose which links he or she follows.

b) What is the "hypertext system" of the "Choose Your Own Adventure" series?

Answer: The book chapters are the hypertext nodes. *The instructions for moving between the chapters are the* hypertext links. *The entire book is the* hypertext system.

1.3.2 ANSWERS

a) What was the Memex system intended to be?

Answer: In 1945, Vannevar Bush, the Science Advisor to President Roosevelt during World War II, proposed Memex (Memory Extension) in a famous article in The Atlantic Monthly. *It was intended to be " . . . a device in which an individual stores his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility." Memex was never actually implemented.*

b) List and discuss existing hypertext systems.

Answer: The World Wide Web is perhaps the most widely used and well-known hypertext system. Because it is an open standard, it has been implemented in one form or another on most every modern operating system. In addition to applications that are specifically designed to read and display the hypertext system of the WWW (HTML), there are a number of other types of applications that support this system, such as world processors, spreadsheets, e-mail applications, and so forth.

Another common system is the online help system used by Microsoft Windows. Although it enables quick access to important reference information, it is proprietary to Microsoft and not implemented on other operating systems.

Dynatext is similar to the help system used by MS-Windows and implemented by many vendors for both Windows-based and Unix-based applications. Unlike the help system in MS-Windows, Dynatext allows a single source of information, regardless of the client system.

1.3.3 ANSWER

a) Compare the Web's "hypertextual" features (e.g., node and link definition, navigation, and so forth) with some of its lesser known predecessors and peers such as Xanadu, Zog, or Hyper-G.

Answer: Some of the lesser known hypertext systems actually are "more true" to hypertext concepts than WWW. The hypertext system provided by HTML (and therefore the WWW) provides links in only a specific direction (from the reference to the referent). These other systems typically supported bidirectional links, "rich links" (links contain information about their referent), and link management systems.

LAB 1.3 SELF-REVIEW QUESTIONS

In order to test your progress, you should be able to answer the following questions.

1) Nodes are the endpoints in a chain of relationships.

a) _____ True
b) _____ False

2) Links represent a relationship between two or more nodes.

a) _____ True
b) _____ False

3) On the WWW, links are bidirectional connections between nodes.

a) _____ True
b) _____ False

4) "Typed" links provide some information about the relationship between the nodes.

a) _____ True
b) _____ False

5) Which of the following is not normally considered part of a computer-based hypertext system?

a) _____ A user-interface that assists the user to navigate and access the information.
b) _____ A tool to author and manage the hypertext system.
c) _____ An SQL database containing the information displayed by the system.
d) _____ Traditional information retrieval mechanisms such as keyword searches.

Quiz answers appear in the Appendix, Section 1.3.

LAB 1.4

THE GOLDEN RULES OF HYPERTEXT

LAB OBJECTIVES

After this lab, you will be able to:

- ✔ Apply the Rules of Hypertext to Hypertext Components
- ✔ Apply the Rules of Hypertext to Different Types of Documents

There are many WWW (and hypertext) evangelists who perceive this medium as the ultimate in online document delivery. Even Ted Nelson's Xanadu was planned as "a *docuverse* (document universe) of all human knowledge." "Put it on the Web" is often viewed as the ultimate online solution for books, documents, art, recordings, and so on.

In reality, not all information is suitably structured for the Web or any other hypertext system. As we discussed earlier, hypertext-appropriate content is that which reflects the way in which people think and learn. Determining whether content is suitable for a hypertext system is summarized in the "Golden Rules of Hypertext," which state that the use of hypertext is most appropriate when:

- The content of the material is a large body of information logically organized and structured into multiple units or fragments.
- These units or fragments are loosely associated with one another, though not necessarily in a sequential manner.
- A user or reader of the material only needs one unit or fragment of the content at any one time.

In the context of a hypertext system, the fragments or units correspond to concepts and nodes. The associations between fragments or units correspond to links. The third rule is more presentational. In the case of the Web, it suggests that a reader only has access to a single Web page at a time given the limitations of a Web browser.

Even if the decision to author a document in hypertext is a good one, poor design of the document can easily present major problems. Just because document content has been defined into fragments and linked does not ensure that it will be effective or attractive.

LAB 1.4 EXERCISES

1.4.1 APPLY THE RULES OF HYPERTEXT TO HYPERTEXT COMPONENTS

The "Golden Rules" primarily address fragments/nodes and usability.

a) What guidelines or restrictions should be presumed regarding use of links? *Hint: For example, does "fragments are loosely associated with one another, though not necessarily in a sequential manner," imply anything about number of or directionality of links?*

1.4.2 APPLY THE RULES OF HYPERTEXT TO DIFFERENT TYPES OF DOCUMENTS

Online/Web-based books have been a major topic of discussion.

a) Given the "Golden Rules of Hypertext," what types of books would be suitable candidates for hypertext and why? Fiction vs. nonfiction? Reference? Educational?

b) Some organizations feel as though all online documentation (e.g., online help systems) should/could be converted to hypertext and made viewable via systems such as the Web. Discuss the appropriateness of such a strategy.

LAB 1.4 EXERCISE ANSWERS

1.4.1 ANSWER

a) What guidelines or restrictions should be presumed regarding use of links?

Answer: The "Golden Rules" imply that all nodes must have (at a minimum) an "entry" and an "exit." This could be accomplished with either two unidirectional links or one bidirectional link. In addition to this minimum, nodes can have any number of additional links, depending upon how closely associated they are with other nodes. This basically means that there must be some way of getting to the node. For example, it would not be very efficient to have Web pages on your site to which there are no links or other ways to access them such as a keyword search.

1.4.2 ANSWERS

a) Given the "Golden Rules of Hypertext," what types of books would be suitable candidates for hypertext? Fiction vs. nonfiction? Reference? Educational?

Answer: Given the sequential nature of fiction, hypertext may only be suitable for fragments that are "visited" sequentially. However, portions of fictional text (e.g., alternate endings) may be better suited for hypertext.

Nonfictional works are good candidates for hypertext, depending upon the required relationships in the content. Reference books are typically very good candidates for hypertext since they would offer a user a broader range of reference and research possibilities.

Like fiction, educational books typically require sequential reading. However, nonsequential structure could allow for various knowledge and skill levels (as in "skip this section if …"). The "fragmentation" of hypertext content can lend itself well to educational material.

b) Some organizations feel as though all online documentation (e.g., online help systems) should/could be converted to hypertext and made viewable via systems such as the Web. Discuss the appropriateness of such a strategy.

Answer: A strategy of converting any documentation to a hypertext system should first involve a careful examination of that documentation and how well its content and structure can be described by "The Golden Rules of Hypertext." For example, an on-line reference system for the company's help desk would be a good candidate for conversion because of the relationship between all of the various pieces of information. On the other hand, a cookbook contains "multiple units or fragments" (the recipes), but there is little relationship between those fragments.

LAB 1.4 SELF-REVIEW QUESTIONS

In order to test your progress, you should be able to answer the following questions.

1) Using the "Golden Rules of Hypertext" helps to ensure that the document is effective as well as attractive.

a) _____ True
b) _____ False

2) Hypertext should be implemented for large documents where linear access is required.

a) _____ True
b) _____ False

3) Hypertext systems are best employed where the information fragments are related in some way.

a) _____ True
b) _____ False

4) Which of the following is not considered one of the "Golden Rules of Hypertext"?

a) _____ The content of the material is logically organized and structured into multiple units or fragments.
b) _____ These units or fragments are loosely associated with one another.
c) _____ Associations between the units or fragments are nonlinear.
d) _____ A reader of the material only needs one unit or fragment of the content at any one time.

Quiz answers appear in the Appendix, Section 1.4.

LAB 1.5

WWW AS A HYPERTEXT SYSTEM

LAB OBJECTIVES

After this lab, you will be able to:

- ✔ Apply Traditional Hypertext Concepts to WWW Content
- ✔ Evaluate WWW as a Hypertext System

While our description of hypertext systems can become very philosophical, we are primarily concerned with the organization and presentation of computer-based information using WWW. As such, a number of conventions are required.

WEB PAGE

A Web page corresponds to our hypertext system node. It is a single file and whatever resources (e.g., graphics, scripts, stylesheets, and so forth) it contains. Thus, it represents a single concept.

WEB DOCUMENT

A Web document is a collection of Web pages and the links between them. Therefore, we think of a Web document as an independent hypertext system. Given this definition, a Web site is a Web document like other information databases and information spaces. Since a Web document is a collection of Web pages, then Web documents can themselves contain other Web documents. That is, they can be compound structures. Using WWW terminology, a Web document is "entered" via such author-defined devices as *home pages, welcome pages, root pages, entry pages,* and *front pages*.

ADDING HYPERTEXT TO WWW CONTENT

Contrary to the conception of many, the World Wide Web is not the first hypertext system—although it is certainly the most successful. Systems with names such as Memex, Xanadu, Augment, and Zog do not have the household familiarity of the WWW. Research into hypertext/hypermedia systems has been going on for more than thirty years. Much of this research has led to tools and techniques that could (and should) be applicable to the Web. Many of what appear to be new problems are actually old problems that have resurfaced.

If you do additional reading on the background of and research in hypertext, you will discover that the WWW is a fairly limited hypertext system. Some of the earlier hypertext systems provided far richer tools for authoring and reading. However, most of these systems would only execute on a single computer running a specific operating system. The greatest strength of the Web is that it is a *distributed* hypertext system capable of linking multiple computers of many types running different operating systems over a network.

Examples of some of the "classical" hypertext features that the Web lacks natively are:

- Bidirectional links—the ability to follow links in both directions.
- Rich links—the ability to determine something (other than just the name or location) about the node a link leads to before activating that link.
- Link management—the ability to manage links such that they always work (i.e., no broken links).
- Publishing and version control—the ability to manage the accuracy or "freshness" of a node.

The WWW has moved some features from the author control to the reader/browser control. For example, the "back" button on a browser logically operates as a backward operating link. But what if the reader is using a browser without a "back" button?

Other hypertext features can be added to Web pages and documents using scripting and programming techniques. Hypertext features such as dropdown and expanding/collapsing menus and inclusion of "rich link" information can be accomplished with client-side scripting and the Document Object Model (DOM).

LAB 1.5 EXERCISES

1.5.1 APPLY TRADITIONAL HYPERTEXT CONCEPTS TO WWW CONTENT

a) Demonstrate how a bidirectional link between two Web pages may be emulated using HTML. *Hint: The "name" attribute of the anchor tag might be one way.*

b) How might you add "rich text" to a Web page?

1.5.2 EVALUATE WWW AS A HYPERTEXT SYSTEM

a) How might the problem of "broken links" on the WWW be addressed with today's technology?

b) Ted Nelson once told a Web conference, "Your future is my past." The implication was that the features he described for his Xanadu system should be incorporated into the WWW. Research some of Xanadu's features (e.g., "transclusion") and discuss how they might be a part of the next generation of WWW.

LAB 1.5 EXERCISE ANSWERS

1.5.1 ANSWERS

a) Demonstrate how a bidirectional link between two Web pages may be emulated using HTML.

Answer: The following example illustrates the use of a "quasi-bidirectional" hyperlink between two pages. Use of the "Go Forward" and "Go Back" links on the respective pages enable a reader to "toggle" between the same locations (using the anchor ***name*** *attribute) on the two pages.*

```
<HTML>
<HEAD>
<TITLE>Example of Bidirectional Link</TITLE>
</HEAD>
<BODY>
<H1>Example of Bidirectional Link</H1>
<A name= "Top" href="...End_2.html#Bottom">Go Forward</A>
</BODY>
</HTML>
```

```
<HTML>
<HEAD>
<TITLE>Example of Bidirectional Link</TITLE>
</HEAD>
<BODY>
<H1>Example of Bidirectional Link</H1>
<A name= "Bottom" href="...End_1.html#Top">Go Back</A>
</BODY>
</HTML>
```


b) How might you add "rich text" to a Web page?

Answer: Using the "onmouseover" attribute to many HTML tags, you could provide a small popup or text in the status bar of the browser that provides addition information about the link.

1.5.2 ANSWERS

a) How might the problem of "broken links" on the WWW be addressed with today's technology?

Answer: For links to pages internal to your Web server, any scripting language that can traverse the directory tree and can file contents could be used. Each file is checked for HREF tags and then a check is made to see if the file is still where it supposed to be.

Some scripting languages like Perl even provide the ability to directly connect to Web servers and can therefore be used to check connectivity. Pages on your Web site could be checked for external URLs using Perl, and then Perl would check the connectivity to that page. Commercial products such as LinkScan will do this for you.

There are also a number of commericial Web development products (i.e., Alliare Home Site, Netobjects Fusion) that have site management functionality.

In addition, numerous technologies are under development that address the problem of "broken links"—link management. Research the work currently being done with URNs (Universal Resource Names) and Persistent URLs (PURLs). The following URLs can provide useful resources:

URNs: http://www.w3.org/Addressing/URL/uri-spec.html
PURLs: http://purl.oclc.org/

b) Research some of Xanadu's features (e.g., "transclusion") and discuss how they might be a part of the next generation of WWW.

Answer: The following URL provides an interesting overview of the Xanadu system: http://www.internetvalley.com/intvalxan.html

LAB 1.5 SELF-REVIEW QUESTIONS

In order to test your progress, you should be able to answer the following questions.

1) A page is node within the hypertext system of the WWW.

a) _____ True
b) _____ False

2) All hypertext systems are distributed.

a) _____ True
b) _____ False

3) The WWW is not a true hypertext system because it does not provide for an author defined entry point.

a) _____ True
b) _____ False

4) Which of the following is not an example of a "classical" hypertext features that the WWW currently lacks?

a) _____ Rich links—the ability to determine something (other than just the name or location) about the node a link leads to before activating that link
b) _____ Native multimedia support
c) _____ Link management—the ability to manage links such that they always work (i.e., no broken links)
d) _____ Publishing and version control—the ability to manage the accuracy or "freshness" of a node

5) Which of the following is one of the key strengths the WWW has over "traditional" hypertext systems?

a) _____ Secure communication
b) _____ Inherently distributed information sources
c) _____ Support for a wide variety of operating systems
d) _____ Multiple language support

6) Which of the following cannot be emulated using HTML or client-side scripting?

a) _____ Bidirectional links
b) _____ Version control
c) _____ Link management
d) _____ Rich links

Quiz answers appear in the Appendix, Section 1.5.

LAB 1.6

HYPERTEXT DOCUMENT ENGINEERING

People have very little patience with poorly designed WWW sites. As one user put it: "The more well-organized a page/site is, the more faith I will have in the information."

—Jakob Neilsen

LAB OBJECTIVES

After this lab, you will be able to:

- ✔ Define the Role of the Hypertext Document Engineer
- ✔ Apply HDE Techniques

Hypertext Document Engineering (HDE) is defined as the application of software design engineering (SDE) techniques to the design of hypertext/Web pages and documents. This is important and relevant because, at some level, the creation of Web pages and documents is similar to the design of a software application. For example:

- Both disciplines are likely to involve the coordination and interoperation of multiple parts that may or may not have been developed by the same person or team.
- Critical to both efforts is the development of interfaces between component parts and users.
- Both disciplines are concerned about the efficiency and reliability of their component parts.

- HDE and SDE are both interested in issues such as version control and maintenance.

At a time when potential Web authors and information providers are told "teach yourself Web publishing in a week," why is Hypertext Document Engineering important? The simplicity of HTML often leads to spontaneous ("markup as you go") page and document design. From a software perspective, many of these pages are analogous to the "spaghetti code" produced by inexperienced programmers.

As has been discussed in previous labs, HDE also incorporates techniques unique to the support and development of hypertext documents. Many of these techniques have evolved from hypertext systems research and experience. In addition, HDE draws from research and experience in human-computer interaction (HCI) and online document authoring. Chapter 5 of this book focuses on HCI technology and the Web.

SDE AND HDE

Software engineering techniques facilitate the design of large and complex software projects. Components of these projects are often independently designed and coded modules produced by different programming staffs. A precise specification of how these modules are to function and their interfaces is what makes "engineered" software projects successful. The most obvious goal of software engineering is that the software produced meets the stated requirements. Four properties that are sufficiently general to be accepted as goals for the entire discipline of software engineering are:

- Modifiability—Is the software designed to be readily (not necessarily easily) modified? Can bug fixes and new features be efficiently added? Can the application evolve to meet new demands?
- Efficiency—Does the software execute efficiently? Is its response time acceptable? Does it make the best use of the resources it requires?
- Reliability—Is the software reliable? Does it fail often or give mysterious errors? Can it crash the computer on which it's executing?
- Understandability—Can the users of the software readily understand how to use it and how it was designed to be used?

In this list of SDE goals, try replacing the words "software" and "application" with "Web page" and/or "Web document" and you will see that the issues are

still applicable. There are, however, some important areas of involvement that make HDE quite distinct from SDE.

HDE requires additional skills of software design and programming. For example, the authoring process is a key component in the design of hypertext content. Likewise, the design of a hypertext document usually involves capturing and organizing a complex body of information and making that body of information accessible to readers/users. The scope of content is usually broader for developers of hypertext-based content. For example, the decision to include multimedia content presents special challenges.

Many Web authors would immediately have little patience with the suggestion to add an engineering paradigm, such as HDE suggests, to their authoring process. After all, publishing on the Web is supposed to be easy. Learning HTML is supposed to be easy.

In reality, HDE helps to address issues that no Web author can easily ignore. Such issues include:

- Page sizing—How big should a Web page be?
- User disorientation—How can a page author help keep a user/reader from feeling "lost in cyberspace"?
- User cognitive overload—How much information can a page realistically deliver to a reader/user without overwhelming him or her?
- Broken links—What Web site designer wants a visitor to encounter broken links (the dreaded error 404 message)?
- Dead-end pages—Transferring legacy information to the Web often leads to pages that contain no links (dead-end pages). A user/reader therefore must use browser navigation tools (e.g., the "back" button) in order to escape the page.

Who of us have not visited Web sites where we encountered such problems? Too often the problems could have been eliminated with careful page and document engineering.

In Lab 1.7 you will be introduced to three design methodologies that address the goals of HDE.

LAB 1.6 EXERCISES

1.6.1 DEFINE THE ROLE OF THE HYPERTEXT DOCUMENT ENGINEER

a) What is the relationship between the "hypertext document engineer" and the content developer/author?

b) What are some inherent problems with the "markup as you go" method of Web page development?

1.6.2 APPLY HDE TECHNIQUES

a) How might a Web page designer address the problem of "dead-end" pages during the development stages?

b) How can a Web page designer help keep a user/reader from feeling "lost in cyberspace"?

c) How can a Web page designer help prevent "information overload"?

d) Like printed media, a Web page design is concerned with how the material looks on the page. How might a Web page designer determine how big a page should be?

LAB 1.6 EXERCISE ANSWERS

1.6.1 ANSWERS

a) What is the relationship between the "hypertext document engineer" and the content developer/author?

Answer: The "hypertext document engineer" designs and maintains the structure of a hypertext document, which usually contains multiple hypertext pages. The hypertext pages that comprise a hypertext document are often written by multiple content developers/authors.

b) What are some inherent problems with the "markup as you go" method of Web page development?

Answer: Links can end up pointing to nowhere, pages may not be accessible from other pages, and it is extremely easy to lose track of the overall structure of the system. As with software development and "spaghetti code," you can end up with "spaghetti pages."

1.6.2 ANSWERS

a) How might a Web page designer address the problem of "dead-end" pages during the development stages?

Answer: Many commercial Web management products have the ability to create templates that can then be applied to each page. The template would include some kind of navigational aid such as a toolbar with links to specific pages or at the very least a link to the home page.

On some sites, I have utilized the Web server's ability to include specific HTML code at specific locations (such as the top and bottom of each page). I can then include a

navigation bar that I can change as I need to without having to edit each file individually. An excellent reference on developing your site with these techniques is The UNIX Web Server Administrator's Interactive Workbook *by James Mohr, also from Prentice Hall.*

b) How can a Web page designer help keep a user/reader from feeling "lost in cyberspace"?

Answer: One important characteristic of any Web site is a consistent look and feel. If a Web site were constantly changing format between pages, you would quickly become just as disoriented as with a book that changes format between pages.

Although some designers like to try out "cool" things on their sites, that tends to be the exception. Look at some of the large Web sites like amazon.com or cnn.com. There are tens of thousands of pages, all with the same look and feel. No matter how deep you have clicked or how long you have spent on each site, you know you are still on the same site.

c) How can a Web page designer help prevent "information overload"?

Answer: The simplest way is to break the material into fragments that address as few topics as possible. Unlike a book, which requires the material to be physically connected, Web documents can contain information spread across multiple pages. As we mentioned previously, this is one of the key benefits of any hypertext system.

Another way is to limit how "cool" your site is. Fancy fonts and spinning graphics might be fine for the MTV Web site, but not for most. A large number of different fonts and "loud" graphics detract from the content.

d) Like printed media, a Web page design is concerned with how the material looks on the page. How might a Web page designer determine how big a page should be?

Answer: Some of the commonly used methods for establishing the size of Web pages are:

1) *The amount of information on a "paper" page*
2) *The size of a hypertext "fragment"*
3) *The amount of information that can "fit" into a particular browser/client window on a certain size terminal screen (e.g., 640 x 480 pixels)*
4) *The amount of information that be downloaded from a server within a specific time period at a particular bandwidth*

LAB 1.6 SELF-REVIEW QUESTIONS

In order to test your progress, you should be able to answer the following questions.

1) Although a Web page is typically not going to cause your browser to crash, Web page designers are just as concerned with "reliability" as software developers.

a) _____ True
b) _____ False

2) For Web page designed "modifiability" simply means being able to replace or update single pages or even groups of pages.

a) _____ True
b) _____ False

3) Web browsers take care of the "human-computer interaction" aspects of Web pages and, therefore, this is not a concern for Web page designers.

a) _____ True
b) _____ False

4) Which of the following is NOT a similiarity between HDE and SDE?

a) _____ Both disciplines are likely to involve the coordination and interoperation of multiple parts that may or may not have been developed by the same person or team.
b) _____ Critical to both efforts is the development of interfaces between component parts and users.
c) _____ Both disciplines are concerned with design which must be able to provide nonlinear access to each component.
d) _____ HDE and SDE are both interested in issues such as version control and maintenance.

Quiz answers appear in the Appendix, Section 1.6.

LAB 1.7

HYPERTEXT DESIGN METHODOLOGIES

LAB OBJECTIVES

After this lab, you will be able to:

- ✔ Analyze and Define Structural Design Issues
- ✔ Redefine a Structure

In this lab, we briefly examine three hypertext design methodologies that can contribute to Hypertext Document Engineering. Each of these methodologies emphasizes the importance of Web/hypertext document usability. In each, document navigation plays a key role since one of the most important factors in the success of a document is the ease with which a reader/user is able to navigate within it, following links between the pages that compose it. Each methodology also pays attention to a user's perception of how the information within the document is organized. No one of these methodologies should be interpreted as being preferable to another. It is the decision of you as the Web author and designer to decide which is the most appropriate for your application.

Each of the three hypertext design methodologies discussed has a different focus:

- The *structure-based* methodology places emphasis on the overall structure of a hypertext document and the implied relationships between nodes/pages presented by that structure.
- The *relationship-based* methodology analyzes the logical relationships between the nodes/pages of a hypertext document and suggests the document design accordingly.

- The *information-based* methodology concentrates on the content fragmenting and organization of a hypertext document. The node/page relationships are a function of the anticipated use of the document.

One thing that you will quickly realize as we discuss these methodologies is that the popular "free-form" architecture of the Web goes away. We no longer feel free to put arbitrary links between any two pages. All Webs, no matter how complex they may appear on the surface, in fact have a well-designed structure underneath.

Each of the methodologies discussed assumes a Web/hypertext document development cycle such as the following Figure 1.2 (adapted from Isakowitz, Stohr, & Balasubramanian).

The *Feasibility* step in this development cycle would likely result in the generation of a feasibility document (probably not a hypertext document). In this document, issues such as user needs and objectives would need to be addressed. The

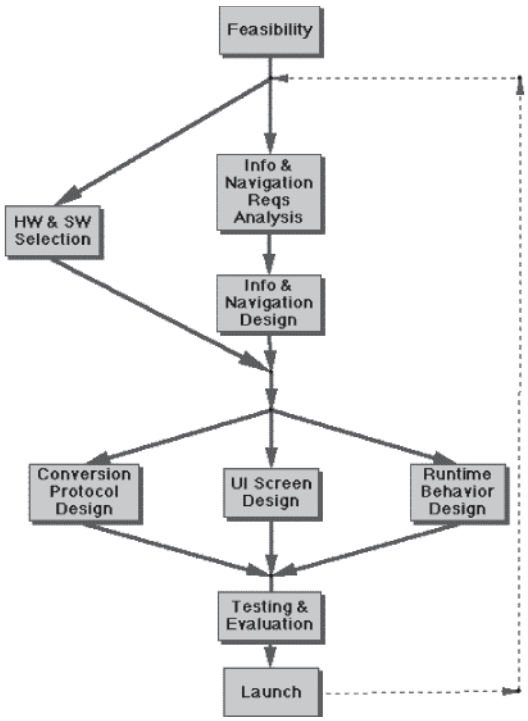

Figure 1.2 ■ A Web/Hypertext Document Development Cycle

Feasibility step would be followed by an *Information/Navigation Requirements Analysis* leading to the development of a requirements document. It is from the requirements document that the document designer/author works.

Each of the methodologies (or perhaps some combination) described in this lab are applicable in the *Information/Navigation Design* phase. It is in this phase that the actual structure of and relationships within the Web/hypertext document begin to take shape. It can generally be assumed that page/document content development occurs independently of this procedure. While one of the methodologies described is information-based, it does not address issues of content quality and quantity.

In the *Conversion Protocol Design* step, each element resulting from the Information/Navigation Design step is coded into the relevant format. For Web documents, this would usually indicate expressing the relationship with HTML.

User Interface Design involves the design of the "look and feel" of each element appearing in the Information/Navigation Design model. User interface design would likely include the descriptions of buttons, content layout, indices, and the page location of navigational tools (e.g., menubars and the like). Topics on User Interface Design are discussed in Chapters 2 through 6, the human-computer interface chapters of this book.

Decisions about how linking and navigational mechanisms are to be implemented are made in the *Runtime Behavior Design* step. It is also at this time when the page author considers the runtime impact of static versus dynamic pages.

THE INFORMATION-STRUCTURE APPROACH

LAB 1.7

Successful hypertext, just as any successful writing project, depends on good design of the contents. The hypertext author who creates a new work or the hypertext editor who takes existing materials and puts them into hypertext form must take great care to produce excellence. The designer who assumes that it is safe to throw everything into the hypertext network and let the reader sort it out will be surprised by the negative reactions.

—Schneiderman & Kearsley, 1989

One thing that does bother me, however, is the belief that hypertext will save the author from having to put material in linear order. Wrong. To think this is to allow for sloppiness in writing and presentation. It is hard

> *work to organize material, but that effort on the part of the writer is essential for the ease of the reader. Take away the need for this discipline and I fear that you may pass the burden on to the reader, who may not be able to cope, and may not care to try. The advent of hypertext is apt to make writing much more difficult, not easier—good writing, that is.*
>
> —Norman, 1988

Even if the decision to author a document in hypertext is a good one, poor design of the document can easily present major problems. Just because document content has been defined in terms of pages and links does not guarantee that the document will be effective or attractive. The potential positive and negative impact of creating a hypertext document is realized when you attempt to specify the relationships between the pages of that document.

The *information-structure approach* assumes that there are four common information organization structures. All other more complex information structures/Webs that we design and use should actually be composed of these four structures. This might also be thought of as a building block approach where all of our beautiful structures and creations are composed of a few fundamental blocks.

The four fundamental information structures are:

1. Sequence
2. Grid
3. Hierarchy
4. Web

These four fundamental structures are distinguished by how difficult they are to understand by a user/reader and how difficult they are to use by an information/content author.

Figure 1.3 is an adaptation from Brockmann, Horton, and Brock (1989). It examines the powers and risks associated with these four structures. The rectangles in this figure represent elements/pages within the structure of a document. In general, these structures represent an information space, a generalization of the hypertext document concept. An information space resembles an information database structured according to content and anticipated use. The *y*-axis represents a relative measure of complexity from the perspective of the user/reader. The *x*-axis represents a relative measure of authoring complexity. Note that this figure is dated 1989, which means that its use of the term "Web" to describe a complex, hypertexted information structure actually predates the World Wide Web.

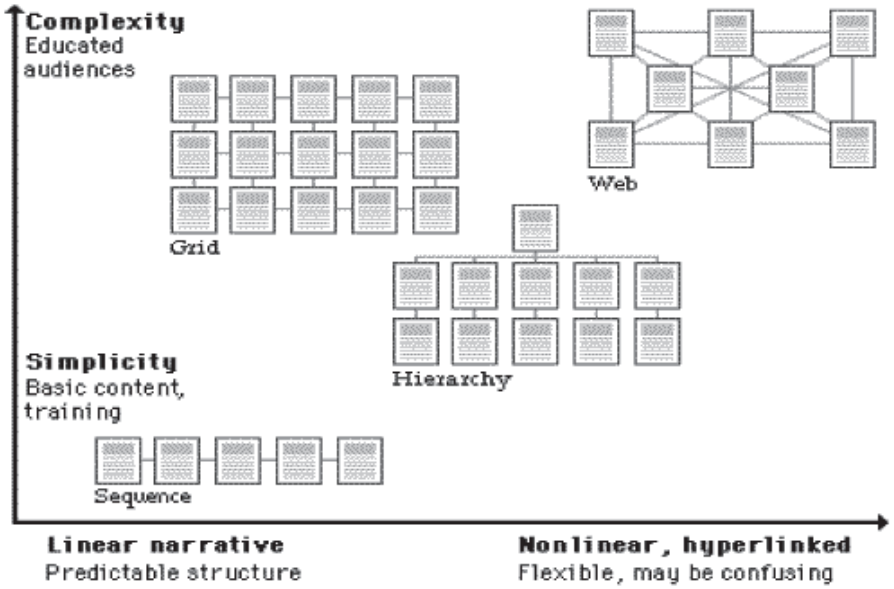

Figure 1.3 ■ Complexity and Predictability of the Four Fundamental Information Structures

Figure 1.3 therefore suggests that the *sequence* structure is the least complex from both the user/reader's and the author's perspective. It also indicates that the content in the sequence structure is more linear (rather than hypertextual) in nature. This figure also suggests that the *web* structure is the more potentially complex to both the user/reader and the author. In terms of the expressive power available to an author, the web structure offers far more than the sequence structure.

We will now discuss each of these structures in detail.

LAB 1.7

THE SEQUENCE STRUCTURE

The simplest of these information structures to design and navigate through is the sequence, because it most closely resembles a conventional paper document. The sequence structure is shown in Figure 1.4. One of the positives ("pros") of this structure is that it is predictable. Users/readers of a document structured in this manner are presented with a familiar and comfortable model.

It is characteristic of the sequence that navigation between nodes can be unidirectional or bidirectional and only to specific (adjacent) nodes. The structure assumes a maximum of two links per node (only one at the ends of the sequence) and allows for simple navigation such as "next," "previous," "forward," and

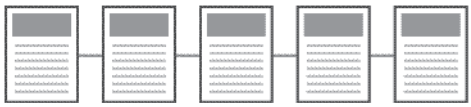

Figure 1.4 ■ The Sequence Information Structure

"back." In this way its navigational functionality is comparable to that of a browser. The structure can only be "entered" at the ends. Therefore, it is unlikely (if not impossible) for a user/reader to become "lost" while reading a document of this form.

Documents structured in this manner are the least likely candidates for hypertext. There is typically not a "rich" relationship between the content of the pages. Except for the navigational links, each of the pages would actually be a "dead end" (i.e., containing no links) page. The sequence structure would best be used for content that is linear in nature, therefore requiring that pages/nodes be read in a very strict sequence. Content that might be suitably organized as a sequence would include online books, online tutorials and courses, or tours. Also, linear documents "converted" to hypertext (modified by the addition of page navigation links—previous page, next page, and so forth) are examples of a use of the sequence structure.

THE GRID STRUCTURE

The *grid structure* is the first of the information structures that can be thought of as multidimensional. The grid structure is shown in Figure 1.5. Therefore, it can be used to define significantly richer relationships between nodes/pages than is possible with the sequence structure. In general, the grid contains pages whose relationships are best described in a tabular fashion.

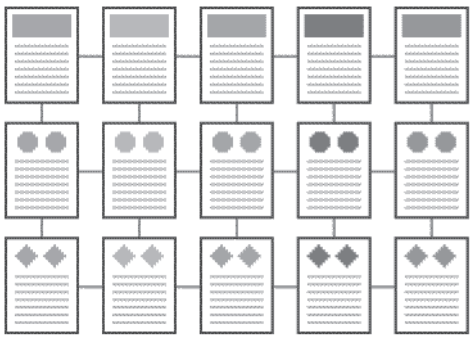

Figure 1.5 ■ The Grid Information Structure

Navigation through a grid structure can be unidirectional or bidirectional and only to specific pages/nodes. Pages/nodes can have two, three, or four links depending upon their location in the grid.

THE HIERARCHICAL STRUCTURE

Use of a *hierarchical* (or tree) structure in a hypertext document allows nodes/pages to be thought of in a hierarchical fashion. The hierarchical structure is shown in Figure 1.6. Like the grid structure, navigation between the pages is best defined multidimensionally.

The hierarchy is perhaps the most common structure for documents that are written modularly. A frequently used example of a document with a hierarchy structure is one that is navigated according to its table of contents. "Home pages" are often designed hierarchically with a tree structure. The document in Figure 1.7 is designed as a tree with a "home page" as the root node, the first level nodes as "submenus," and all remaining nodes consisting of individual pages or documents.

Along with the sequence structure, the *hierarchical structure* is probably the most generally familiar. Its use ranges from general taxonomies, genealogical charts, and organizational charts to computer file systems. The allowable navigational paths are very well defined within this structure. Navigation can be unidirectional or bidirectional and only to specific nodes/pages.

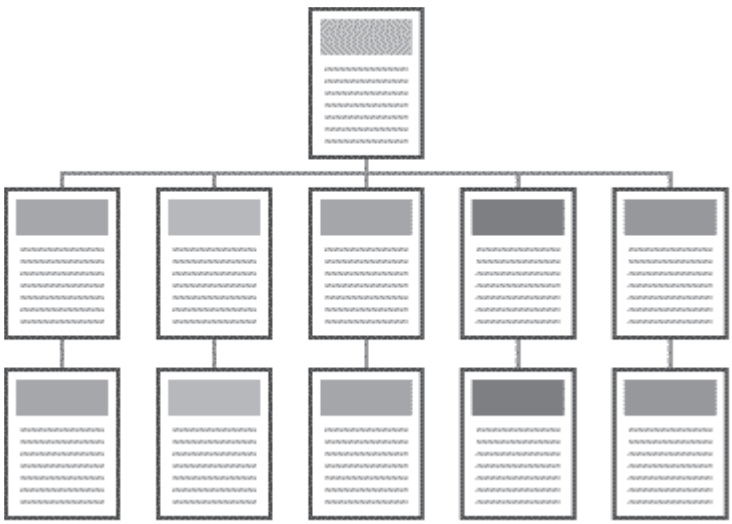

Figure 1.6 ■ The Hierarchical Information Structure

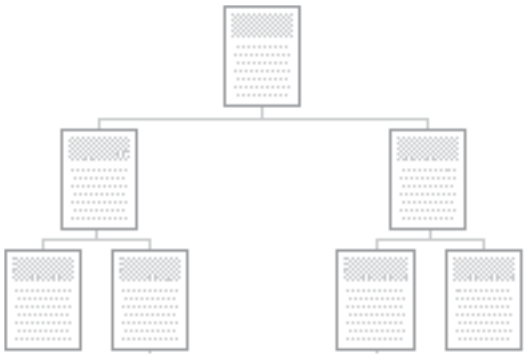

Figure 1.7 ■ A Hierarchical Web Document

The hierarchical structure is more expressive than the grid structure due to the permissible number of links per node/page. Based upon the complexity of the structure and how many nodes, subnodes, and so on the hierarchy contains, the number of links per node is basically unlimited.

There are numerous examples of defining a hierarchical information structure. In the chapter entitled "Supporting Web Servers" of Volume 2 of this series, mapping a file system taxonomy to a hierarchical information structure is described.

PROBLEMS WITH THE HIERARCHICAL STRUCTURE There are well-known problems associated with hierarchical information structures that can lead to navigation and/or information overloading issues.

One of these problems is a hierarchy that is too deep. This means that a reader/user has to follow too long a path in order to get to the desired information. This condition is sometimes described by saying that a user must "drill too deep" or that the desired information requires "too many clicks." The fear to an author should be that a reader/user gets tired and/or frustrated and just quits. Figure 1.8 illustrates a hierarchical structure that is too deep (e.g., five clicks from the root to some pages).

Similarly, hierarchical structures can be described as "too shallow." In Figure 1.9, too many nodes/pages have been "pushed" to the top of the hierarchy. As a result, a user/reader is faced with too many choices when entering the structure (in this case, 9). Later is this lab we will propose that nine items is just too many for most people to remember.

THE WEB STRUCTURE

The *web* is, of course, the most expressive structure from a hypertextual perspective. It may also be the most misunderstood among Web authors. The web structure is shown in Figure 1.10.

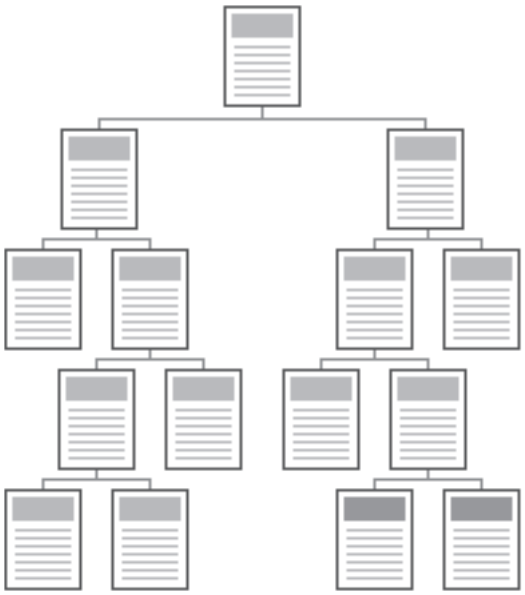

Figure 1.8 ■ A "Too-Deep" Hierarchy

Random links in webs are comparable to GOTOs in programs.

—Tomas Isakowitz

Structured programming techniques condemn the use of transfer statements (GOTO statements) in computer programs. Programs containing excessive GOTOs are often referred to as "spaghetti code." Programming languages such as Pascal were designed to support Structured Programming—software development using well-defined control structures logically eliminating the need for GOTOs. Random links in webs lead to "spaghetti webs."

Figure 1.9 ■ A "Too-Shallow" Hierarchy

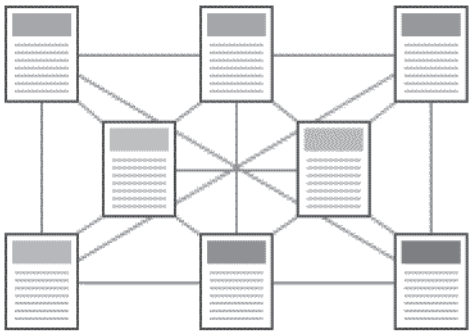

Figure 1.10 ■ The Web Information Structure

By definition the web structure supports unidirectional or bidirectional navigation between nodes/pages connected with links. It also allows pages/nodes to have any number of nodes.

THE RELATIONSHIP MANAGEMENT METHODOLOGY APPROACH

The Relationship Management Methodology (RMM) approach is an example of a relationship-based design methodology. It was invented by Isakowitz, Stohr, and Balasubramanian. It is based upon an *entity-relationship* paradigm.

The definitive description of RMM can be found in "RMM: A Methodology for Structured Hypermedia Design." Communications of the ACM 38 *(August 1995).*

Entity–relationship (ER) modeling is an analysis technique usually associated with database design and object-oriented programming (OOP). Systems are described in terms of the entities that compose them (nodes in a hypertext system) and the relationships between these entities (links or logical groups of nodes in a hypertext system). More specifically, *entities* are system object/components defined in terms of the roles they play in a specific system. *Relationships* are named associations between two or more entities.

ER diagrams are a stylized technique for specifying entities and relationships in a specific system. Entities are represented by rectangles, relationships by diamonds. The following ER diagram, Figure 1.11, describes the entities and relationships in a typical organizational structure.

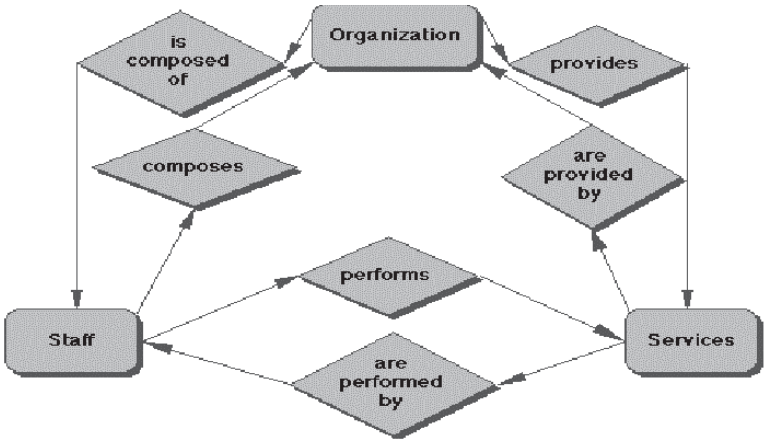

Figure 1.11 ■ An Organization/Staff/Services Entity–Relationship Diagram

If we assume that the entities in this diagram (Organization, Staff, and Services) are hypertext pages or documents, then it is only necessary to define how we express the defined relationships between them. This description of an organization is far richer and more descriptive than one that might be described hierarchically (as is often the case in an organization chart).

RMM defines three fundamental structures for describing relationships between entities/nodes. They are:

- Index—a table of contents to a list of entity instances, providing direct access to each listed item.
- Guided Tour—a linear path through a collection of entities/nodes that permits forward and backward motion along the path.
- Indexed Guided Tour—a combination of the features of an index and a guided tour.

Let's describe these structures by example. Suppose that the "Organization" that we are describing is a part of the portion of the White House personnel of the U.S. government. The relevant "Personnel" is the President, the Vice President, and the First Lady (Bill Clinton, Al Gore, and Hillary Clinton at the time this chapter was written).

Use of an *Index* to describe the Organization-Staff relationship might look like Figure 1.12.

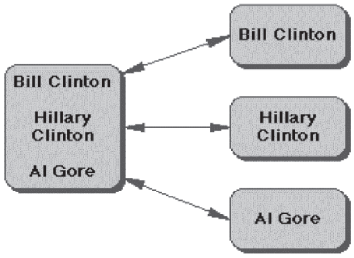

Figure 1.12 ■ Organization/Personnel Relationships Expressed as an Index

The "Organization" node (the leftmost entity) controls as a menu all access to the "Personnel" nodes. It is impossible to go from one "Personnel" node (e.g., Bill Clinton) to another (e.g., Al Gore) without going through the menu. Defining this structure in terms of HTML markup is a trivial exercise.

If the Web site designer chose to use the *Guided Tour* structure to describe this relationship, it might look like Figure 1.13.

Choice of this structure means that users/readers can learn about the "Personnel" only in a specific order. The tour begins with Bill Clinton and ends with Al Gore. However, once within the tour, the user/reader may be able to move (in this case) forward and backward. Once again, defining this structure in HTML is quite easy.

A discussion of the Indexed Guided Tour is left as an exercise.

In summary, using the RMM approach to design a hypertext system/Web site requires:

1. Determination of the core concepts/entities that make up the system; these entities can be simple pages/nodes or more complex structures.
2. Use of an entity-relationship diagram (ERD) to define the relationships between these entities/nodes.
3. Determination of the most appropriate structures (Index, Guided Tour, or Indexed Guided Tour) to express the relationships within the ERD.
4. Implementation.

As is often the case in many design disciplines, Steps 1, 2, and 3 are the most difficult and challenging. Since the relationship structures can be essentially thought of as templates, careful design makes Step 4 quite straightforward.

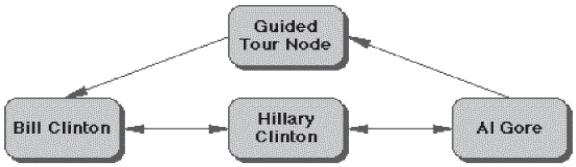

Figure 1.13 ■ Organization/Personnel Relationships Expressed as a Guided Tour

THE INFORMATION-MAPPING APPROACH

In our earlier discussion of "The Golden Rules of Hypertext," we stressed the importance of breaking information up into independent, usable fragments. It was asserted that bodies of information that could be "broken up" in this way were good candidates for hypertext systems (i.e., the Web). A larger question then becomes, how big should these information fragments be?

One issue that web designers/authors have struggled with since the beginning is "How big should a page be?" It's clear that users/readers have difficulty with long pages through which they are forced to scroll. It's also clear that small, short pages can create a maintenance nightmare for authors/designers. The *information-mapping approach* seeks to provide a logical answer to these and other questions.

The information-mapping approach was invented by Robert Horn. In the early days (pre-Web) of online documentation systems, Horn realized that use of a "paper metaphor" on computer systems was not effective.

In paper systems, information is defined in terms of sentences, paragraphs, pages, and so on. While this structure has been very effective for paper-based media (e.g, books, newspapers, and such), Horn believed that it is not necessarily the best model for information on computers. His approach was largely substantiated by the psychology of how humans obtain and retain information.

Paper documents are defined by physical size. Pages typically have a fixed size based upon the type of document of which they are a part (e.g., the page size in a legal document is different than that of a paperback book). Unlike their paper counterparts, hypertext pages have no inherently defined limitations on their size. Page size may be considered an unfortunate consequence of hypertext pages adopted by the author.

There are two strategies that can be adopted by a hypertext author and page designer in making decisions related to page sizing. These strategies address the issues of quantity of page content and factors affecting page viewing by a reader. More specifically, these strategies dictate page sizing based on human short-term memory considerations and viewing area limitations.

Viewing area limitations cannot be immediately addressed by most authors in that they are unlikely to know the wide variety of computer screens, terminals, and other hardware that reader/users are using. This is becoming more of an issue with alternative web agents such as WebTV™ and access via Personal Digital Assistants (PDAs).

SHORT-TERM MEMORY

Short-term memory refers to the facts, concepts, or ideas of which individuals are conscious or that they can recall to consciousness quickly. Short-term memory is sometimes referred to as post-distractional memory, since people can become distracted and forget if their attention is not focused on the contents of that memory.

Pioneering psychological research by Miller and Simon indicated that the capacity of short-term memory is **7 ± 2** chunks of information. "Chunks" of information imply that the information is in some way related, such as in type and form. Miller and Simon's research is consistent with our ability to remember telephone numbers (in the United States, 7 digits) and zip codes (5 digits), but also explains our fears when the telephone company and post office threaten to increase the number of digits in both.

INFORMATION BLOCKS AND INFORMATION MAPS

It was in this short-term memory research that Robert Horn found the method for making the paper to online information transition. He realized that for users reading information online, the major problem was one of retention, not unit size. (For example, after scrolling through a long Web page, can you easily remember content at the top?). The information-mapping approach is the result. In this approach, the paper paragraph is replaced by an *information block,* the paper page is replaced by an *information map.*

An information block is the basic unit of online content. Like the paper paragraph it replaces, it is the smallest unit into which similar related units of information are collected. As might be expected, an information block contains 7 ± 2 chunks of related information, where a chunk might be a sentence, an image, or a table. The relationships between the chunks within the information block are what helps the user/reader to remember them.

An information map contains 7 ± 2 information blocks about a specific topic. Therefore, the information map replaces the paper page as being the basic unit of online content to be provided to a reader/user. According to this design methodology, each Web page/hypertext node should be an information map about a particular subject or concept. If this information map/Web page is correctly

constructed, then it optimizes a user/reader's ability to retain the information within it due to short-term memory considerations.

INFORMATION-MAPPING PRINCIPLES

The information-mapping approach also includes four principles to aid authors/designers in determining how to define "chunks" of information.

- The *chunking principle* requires that all information be grouped into manageable, independent units or "chunks."
- The *relevance principle* directs that each of these chunked units should contain only information that relates to one main point.
- The *consistency principle* states that an author should, for similar subject matters, use consistent vocabulary, labels, formats, and sequences.
- The *labeling principle* specifies that every unit of information and/or groups of information should be labeled according to specific criteria.

HYPERTRAILS

To this point, the information-mapping approach has focused on the definition and construction of hypertext nodes/Web pages. Nothing has been said about how information maps are organized into a hypertext system/Web site.

Robert Horn refers to an organized hypertext system/Web site as a *hypertrail*. In particular, a hypertrail is defined as a set of links between chunks of information (information blocks or information maps) that organize and sequence information about a particular function characteristic of the subject matter.

As with the other design methodologies we have discussed in this chapter, the information-mapping approach defines a limited number of hypertrail types. These types are indicative of how the hypertrail/Web site is to be used/read/navigated by a user/reader. Some of the fundamental types of hypertrails are:

- Prerequisite
- Classification
- Project
- Structure
- Decision
- Example

A number of these types directly correspond to applications we have seen with other methodologies earlier in this chapter. For a complete description of the hypertrail types, you should refer to *Mapping Hypertext,* by Robert Horn.

Figure 1.14 illustrates the use of a prerequisite hypertrail. As the name suggests, such a hypertrail requires that nodes may have prerequisite requirements associated with them. This hypertrail describes an educational curriculum leading to a Web Document Engineering course.

This hypertrail defines the courses and the order in which they may be taken prior to the Web Document Engineering course. It is interpreted as follows:

- Writing Online Documentation, Human-Computer Interaction, and Concepts of Hypertext are all prerequisites for Introduction to HTML, but have no prerequisites themselves and may be taken in parallel.
- Multimedia Design, Introduction to HTML, and Perl Programming can all be taken in parallel.
- Multimedia Design, Introduction to HTML, and CGI Scripting are prerequisites for Advanced HTML.
- Perl Programming is a prerequisite for CGI Scripting.
- Advanced HTML is a prerequisite for Web Document Engineering.

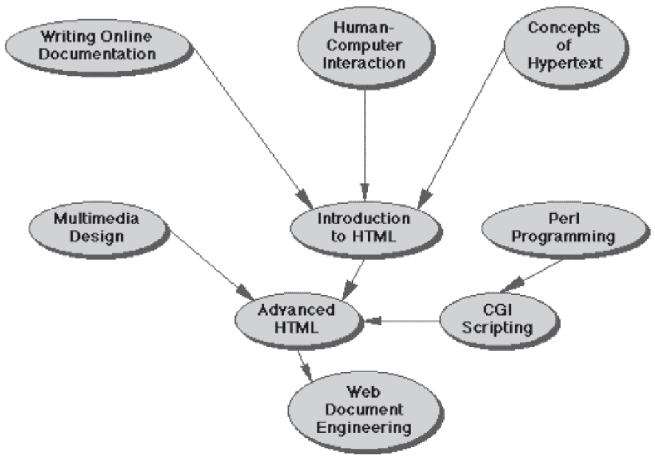

Figure 1.14 ■ Prerequisite Hypertrail Description of a Web Document Engineering Course

The contents of the nodes in this hypertrail are independent of its definition. The nodes could contain descriptions of the courses (as in a course catalog) or could even contain course content (as in an online learning system). The most important characteristic of the hypertrail is the relationships expressed between the nodes and the paths that can be used to navigate the system.

LAB 1.7 EXERCISES

1.7.1 ANALYZE AND DEFINE STRUCTURAL DESIGN ISSUES

Information structures are often combinations of the four fundamental structures. For example, a node within a hierarchical structure might actually be an embedded Web structure.

a) Define a circumstance where this might be the most appropriate structural design.

"Information overload" is human-computer interface (HCI) issue of concern to hypertext document designers. An example of information overload previously given dealt with hierarchical structures that are "too shallow."

b) What is the potential impact of short-term memory on information overload?

c) Discuss the advantages and disadvantages of a hierarchical structure versus a Web structure. (*Note:* Consider the problem of porting existing documents to Web pages.)

d) The size of a "chunk" of information is important to retaining it. Compare the amount of information in each article on the CNN Web site (www.cnn.com) and Time Magazine Web site (http://www.time.com/time/magazine/toc/).

1.7.2 Redefine a Structure

Figure 1.15 is a three-row, two-column grid structure. If the designer of this structure wanted to add an additional column, then navigation from the row-1, column-1 node to the row-1, column-3 node would have to go through the row-1, column-2 node.

a) How could the third column be added such that the row-1, column-3 node was directly accessible from the row-1, column-1, and the row-1, column-2 nodes? *Hint: The rows in this structure are sequence structures. Must rows in a grid structure always be sequence structures?*

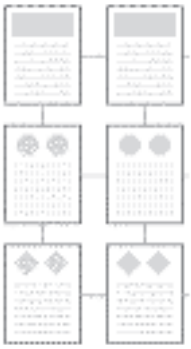

Figure 1.15 ■ A 3 × 2 Grid Structure

b) The Indexed Guided Tour combines the best features of the Index and the Guided Tour structures. What might this structure look like?

Consider the Web page in Figure 1.16 as an example "information map." In this example, it really doesn't matter whether you can read the text, the number of organizational units is the most important feature.

c) Does the Web page in Figure 1.16 meet the conditions of being a good information-mapping approach?

d) What could be some other common hypertrails?

LAB 1.7 EXERCISE ANSWERS

1.7.1 ANSWERS

a) Define a circumstance where this might be the most appropriate structural design.

Answer: It is quite common for a hierarchically designed Web site to have links on the "root page" that extend to the Internet "in general." In such a case, the "Internet/Web" is logically one of the nodes of the hierarchical structure. For example, in the following diagram, Figure 1.17, the "cyberspace" node is the entire World Wide Web.

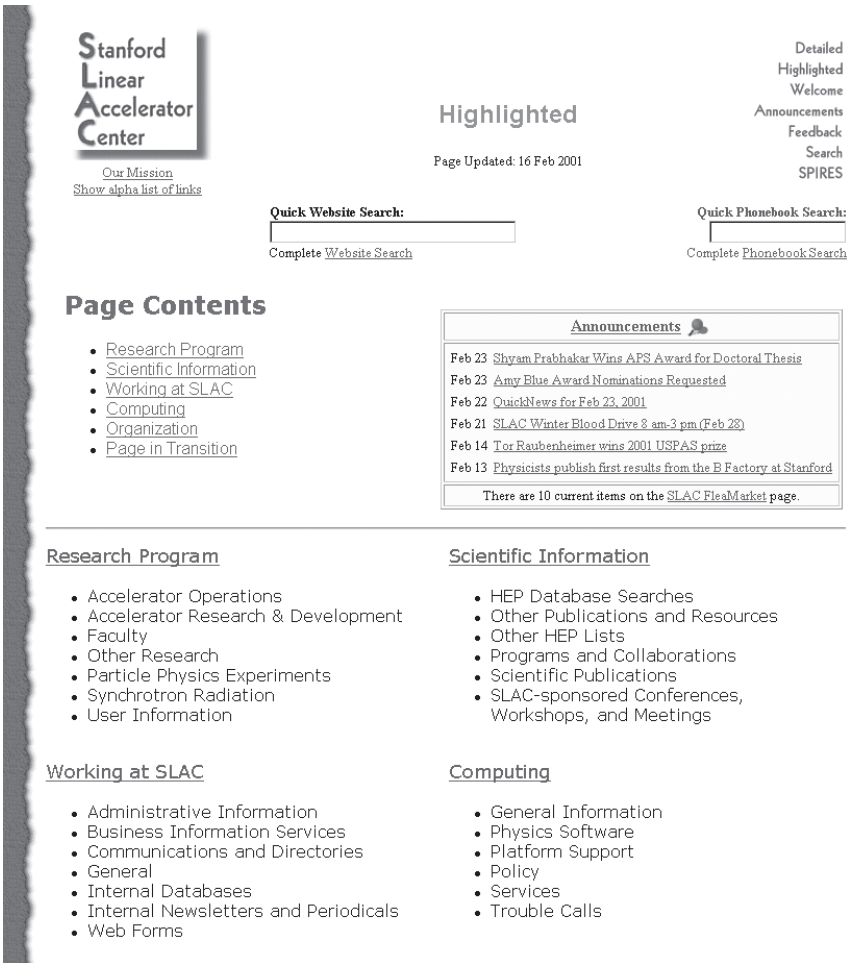

Figure 1.16 ■ A Home Page as an Information Map

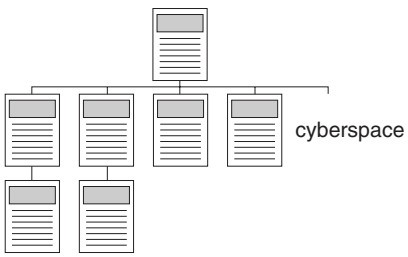

Figure 1.17 ■ A "Composite" Information Structure

b) What is the potential impact of short-term memory on information overload?

Answer: Short-term memory considerations would suggest that hierarchical structures have a maximum of 9 (i.e., 7 + 2) nodes at the first level (that level immediately below the root node). If the number of nodes at that level is greater than 9, a user would have difficulty retaining them in short-term memory (information overload) and the hierarchy would be considered "too shallow."

c) Discuss the advantages and disadvantages of a hierarchical structure versus a Web structure. (Note: Consider the problem of porting existing documents to Web pages.)

Answer: If sets of pages can be easily grouped into their own Web "documents" with little or no relationship to other groups, there is little need to have many links between them. This tends to lead away from a Web structure. In addition, the more the topics themselves are hierarchical (e.g., a company organization), the more you should tend toward a hierarchical structure.

Web-based structures are more suited to information fragments with a lot of interrelationships. On the other hand, if not properly managed, Web structures can get out of hand and may be difficult to navigate.

When porting from paper-based documents to a Web document, a hierarchical structure is quicker to implement as the document is probably already in some kind of hierarchy (chapters, sections, and so forth).

d) The size of a "chunk" of information is important to retaining it. Compare the amount of information in each article on the CNN Web site (www.cnn.com) and Time Magazine Web site (http://www.time.com/time/magazine/toc/).

Answer: Basically all of the articles on the CNN Web site are single pages that provide you with just the basic details of a particular story. There are links to other stories (others about a single incident or about related topics), but each article is short and can be typically read in a few minutes. It is therefore easier to remember most of the article.

On the other hand, the Time Magazine has magazine-length articles (what else?). These are spread across multiple pages and each page usually has more text than a page on the CNN site. Because more information is presented, a smaller percentage of that information is retained.

1.7.2 ANSWERS

a) How could the third column be added such that the row-1, column-3 node was directly accessible from the row-1, column-1, and the row-1, column-2 nodes? *Hint:* The rows in this structure are sequence structures. Must rows in a grid structure always be sequence structures?

Answer: Replace each row (currently a two-node sequence) with a web structure consisting of three nodes in a "ring." The nodes in the ring are aligned appropriately in order to maintain the required column/vertical relationship that exists in the original grid. The two-dimensional grid then becomes a structure more closely resembling a cylinder.

b) The Indexed Guided Tour combines the best features of the Index and the Guided Tour structures. What might this structure look like?

Answer: Using this structure, the Executive Branch "Organization" might look like Figure 1.18.

While the index node/entity on the left provides an entrance to the "Personnel" nodes, a reader/user is free to move about those nodes without always returning to a menu. The relationships between these nodes/pages can be readily expressed using HTML markup, such as creating a set of links at the top or bottom of each page. Although not shown in this graphic, links could be provided between any node (such as between the President and Vice President).

c) Does the Web page in Figure 1.16 meet the conditions of being a good information mapping approach?

Answer: Consider the components of this Web page. As an information map, it consists of seven information blocks:

1. *The logo, title, and date*
2. *The menu bar (containing six buttons)*
3. *A paragraph of text*
4. *Some text within horizontal rules*

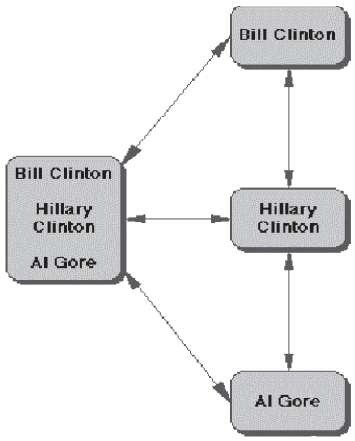

Figure 1.18 ■ Example of Indexed Guided Tour

5. *A bulleted list (containing a sublist); the number of bullets within each list is well within short term memory limitations*
6. *Additional text within horizontal rules*
7. *A bottom menu bar (containing four buttons)*

Even without being able to read the text, we can see that access to the information on the page would be fairly straightforward.

d) What could be some other common hypertrails?

Answer: Other common hypertrails may be as follows:

- *Definition—Links move from one definition to another. An online dictionary would be one example.*
- *Chronological—Links move along a time line. A Web site detailing a historical event would be one example.*
- *Geographic—Links move from one location to another. A company's Web site describing sales areas would be on example.*

In fact, any time you have a list of information that moves somewhat sequentially from one to the other, you have a hypertrail.

LAB 1.7 SELF-REVIEW QUESTIONS

In order to test your progress, you should be able to answer the following questions.

1) Hypertrails are what some browsers call the "History," which is a list of previously visited sites.

a) _____ True
b) _____ False

2) Entity–relationship (ER) modeling only applies to topics where physical objects are involved and therefore does not apply to every Web site.

a) _____ True
b) _____ False

3) A common problem with a hierarchical structure is that they can grow "too deep" and are therefore difficult to navigate.

a) _____ True
b) _____ False

4) A common problem with a hierarchical structure is that they can be "too shallow" and therefore the reader is presented with too many choices.

a) _____ True
b) _____ False

5) Decisions about how linking and navigational mechanisms are part of the User Interface Design phase.

a) _____ True
b) _____ False

6) Which of the following is not a hypertext design methodology?

a) _____ structure-based
b) _____ reference-based
c) _____ relationship-based
d) _____ information-based

7) What is the normally considered the maximum number of objects one can be expected to retain in short-term memory (± 2)?

a) _____ 5
b) _____ 7
c) _____ 9
d) _____ 11

8) The four fundamental information structures are:

a) _____ Structured, Grid, Hierarchy, Web
b) _____ Sequence, Grid, Hierarchy, Web
c) _____ Sequence, Graph, Hierarchy, Web
d) _____ Sequence, Grid, Organization, Web

9) Which are the four information-mapping principles?

a) _____ Chunking, Relationship, Consistency, Labeling
b) _____ Chunking, Relevance, Consistency, Labeling
c) _____ Chunking, Relevance, Consistency, Linking
d) _____ Clarity, Relevance, Consistency, Labeling

Quiz answers appear in the Appendix, Section 1.7.

CHAPTER 1

TEST YOUR THINKING

The projects in this section are meant to allow you to utilize all of the skills that you have acquired throughout this chapter. The answers to these projects can be found at the companion Web site to this book: http://www.phptr.com/phptrinteractive.

Visit the Web site periodically to share and discuss your answers.

1) Expand the Organization/Staff/Services example given in the discussion of the RMM approach into a well-defined hypertext information structure. Justify the use of the relationship structures that you choose.
2) Compare your result in Project 1 with a comparable organizational structure defined hierarchically.
3) The "project" hypertrail is defined as the hypertextual equivalent of the commonly used PERT chart. Create an example of a project hypertrail. (You may have to find out what a PERT chart is.)