
World-Wide Web

A hypermedia information universe

being woven across the global Internet

Awareness report

Hans C. Arents

Materials Information Processing Systems group
Department of Metallurgy and Materials Engineering
K.U.Leuven, W. de Croylaan 2, B-3001 Leuven, Belgium

n Executive summary

This awareness report introduces in detail the World-Wide Web, a hypermedia information universe which is now being woven across the global Internet. It introduces the World-Wide Web itself, its underlying philosophy and design, and briefly discusses Mosaic, at present the most popular browser for accessing and displaying hypermedia documents on the Web. It then discusses in detail HTML, the underlying markup language used for creating and exchanging hypermedia documents, by looking at the elements and functionalities this markup supports. Finally, it looks ahead to the possibilities of HTML+, the proposed extension of HTML which significantly enhances the functionalities of the Web.

n Table of contents

World-Wide Web	2
Goal and use	2
Brief history	3
Data model	3
NCSA Mosaic.....	4
Brief history	4
Functionalities	4
HTML markup.....	6
Design.....	6
Elements.....	6
HTML+ markup.....	10
Design.....	10
Elements.....	10
Glossary	13
Appendix 1: HTML DTD.....	15
Appendix 2: HTML+ DTD.....	19

This report was compiled from information available on the World-Wide Web. For more information on:

- the World-Wide Web: <http://info.cern.ch/hypertext/WWW/TheProject.html>
- the NCSA browser: <http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSAMosaicHome.html>
- HTML and HTML+: <http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html>

For any remarks or questions please contact:

drs. Hans C. Arents
 Materials Information Processing Systems group
 Department of Metallurgy and Materials Engineering
 K.U.Leuven, W. de Croylaan 2, B-3001 Leuven, Belgium

Tel: +32-16-22.09.31 Fax: +32-16-20.79.95 E-mail: hca@mtm.kuleuven.ac.be

n World-Wide Web

For fifty years, people have dreamt of the concept of a universal information universe - data that would not only be accessible to people around the world, but information that would link easily to other pieces of information so that any piece of relevant data would be quickly found by a user. It was in the 1960's when this idea was explored further, giving rise to visions of a "docuverse" that people could swim through, and that would revolutionize all aspects of human-information interaction, particularly in the educational field. Only now has computer and networking technology caught up with these dreams, making it possible to implement such a hypermedia information universe on a global scale.

Goal and use

The official description defines the *World-Wide Web* as a "wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents". What the **World-Wide Web** (*WWW*, pronounced *W3*) project has done is provide users on computer networks with a consistent means of accessing a variety of media in a simplified fashion. Using popular interfaces to the Web such as Mosaic, Cello or tkWWW, the Web project has changed the way people view and create information - it has created the *first true global hypermedia network*.

The operation of the Web relies on *hypertext* as its means of interacting with users. Hypertext is basically the same as regular text - it can be stored, read, searched, or edited - with an important addition: hypertext contains connections within the text to other documents. For instance, suppose you were able to somehow select (with a mouse or with your finger) the word "hypertext" in the sentence before this one. In a hypertext system, you would then have one or more documents related to hypertext appear before you - a history of hypertext, for example, or the Webster Dictionary's definition of hypertext. These new texts would themselves have links and connections to other documents, and continually selecting text would take you on a free-associative tour of information. In this way, hypertext links, called *hyperlinks*, can create a complex virtual web of connections. *Hypermedia* is hypertext with an additional feature: hypermedia documents contain links not only to other pieces of text, but also to other forms of media (sounds, images, and movies). Images themselves can be selected to link to sounds or documents. The Web, although still in its early years, allows all of this (and soon more) in real time. It facilitates the easy exchange of hypermedia information through networked environments from anything as small as two PCs connected together to something as large as the global Internet. The Internet is the catch-all word used to describe the massive world-wide "network of networks" of computers. On any given day it connects roughly 15 million users at over 30 thousand sites in over 50 countries. The World-Wide Web is mostly used on the Internet; they do not mean the same thing however. The Web refers to a body of hypertext and hypermedia information - an abstract, fast growing universe of knowledge, while the Internet refers to the physical side of the global network, a giant mass of cables and computers.

Brief history

The Web began in March 1989, when Tim Berners-Lee of CERN (a collective of European high-energy physics researchers) proposed the project to be used as a means of transporting research and ideas effectively throughout the organization. Effective communications was a goal of CERNs for many years, as its members were located in a number of countries. As an illustration of the Web's growth, from January to August 1993, the amount of network traffic (in bytes) across the National Science Foundation's (NSF's) North American network attributed to Web use multiplied by 414 times. The Web is now ranked 13th of all network services in terms of sheer byte traffic. In January 1993 its rank was 127. Today there are at least 100 hypertext Web servers in use throughout the world. Since its inception, the CERN Web server traffic has *doubled every four months* - twice the rate of Internet expansion.

Data model

The World-Wide Web works under the popular client-server model. A *Web server* is a program running on a computer whose purpose is to serve documents to other computers when asked to. A *Web client* is a program which provides a visual interface to the user and requests documents from a server as the user asks for them. Because the server does a minimal amount of work (retrieving documents) and only operates when a document is requested, it requires only a minimal workload on the computer running it. This client-server model was chosen because the Web must cope with a widely distributed heterogeneous set of computers running different applications which use different preferred data formats.

Here's an example of how the client-server communication process works:

- Running a Web client such as Mosaic (also called a *browser*), the user selects a piece of hypertext connected to another document, e.g. "The History of Computers".
- The Web client connects to a computer specified by a network address somewhere on the Internet and asks that computer's Web server for the document "The History of Computers".
- The server responds by sending the text and any other media within that document (pictures, sounds, or movies) to the users screen.

The World-Wide Web is composed of thousands of these virtual transactions taking place per hour throughout the world, creating a web of massive and continuous information flow.

Future Web servers will include encryption and client authentication abilities - they will be able to send and receive secure data and be more selective as to which clients receive information. This will allow freer communications among Web users and will make sure that sensitive data is kept private. It will be harder to compromise the security of commercial servers and educational servers which wish to keep information local. Improvements in security will facilitate the idea of "*pay-per-view*" *hypermedia*, a concept which many public and private commercial interests are currently actively pursuing.

n NCSA Mosaic

Months after CERN's original proposal, the **N**ational **C**enter for **S**upercomputing **A**pplications (NCSA) began a project to create an interface to the World-Wide Web. One of NCSA's missions is to aid the scientific research community by producing widely available, non-commercial software. Another of its goals is to investigate new research technologies in the hope that commercial interests will be able to profit from them. In these ways, the Web project was quite appropriate. The NCSA's Software Design Group began work on a versatile, multi-platform interface to the World-Wide Web, and called it *Mosaic*.

Brief history

In the first half of 1993, the first version of NCSA's Web browser was made available to the Internet community. Because earlier beta versions were distributed, Mosaic had developed a strong yet small following by the time it was officially released. Because of the number of traditional Internet services it could handle, and due to its easy, point-and-click hypermedia interface, Mosaic soon became the *most popular interface* to the Web. Currently versions of Mosaic can run on Suns, IBM-compatibles running Microsoft Windows, Macintoshes, and computers running various other forms of UNIX.

Functionalities

Mosaic running on every supported computer should have the following basic features:

- A consistent mouse-driven graphical interface.
- Multiple independent document viewing windows.
- The ability to display hypertext and hypermedia documents.
- The ability to make basic hypermedia links to and support most Internet network services.
- The ability to display the ISO 8859 set of languages such as French, German and Cyrillic.
- The ability to display electronic text in a variety of fonts and styles (bold, italic, or strikethrough).
- The ability to display layout elements such as paragraphs, lists, numbered and bulleted lists, ...
- Support for interactive fill-out forms inside documents, to enable powerful database and search engine front-ends. Fill-out forms can contain text entry areas (single- or multi-line), option buttons, radio buttons, option menus, scrolled lists, and image maps.
- Visited document history list per window.
- Global history with previously visited locations; global history is persistent across sessions.
- Hotlist/bookmark list of documents you found to be interesting; hotlist is persistent across sessions.
- Personal annotations which can later be edited or deleted, and hyperlinks to existing annotations are inlined into subsequent accesses of an annotated document. (Any document from any server via any access method can be annotated.)

- Save/mail/print documents in several formats, including formatted ASCII text and PostScript.
- Inlined images (in GIF or XBM format) of up to 256 colors within documents: images can be included anywhere inside a document, and can act as hyperlink anchors. Image files themselves can be located anywhere on the network. Images can act as interactive maps, so clicking on them sends coordinates of click to remote server.
- Automatic dithering of inlined images on monochrome displays.
- Flexible inlined-image caching with customizable image cache size.
- Delayed image loading mode, to avoid automatic loading of all images in accessed documents for users with slow network connections.
- Support for sounds (Macintosh, Sun audio format, ...) and movies (MPEG-1 and QuickTime).
- Full customizability of recognized formats, external viewers, and file extensions.
- Built-in support for recognizing and handling GIF, JPEG, TIFF, audio, AIFF, DVI, MPEG, MIME, XWD, RGB, HDF, PostScript files and forking off appropriate viewers.
- The ability to fire off arbitrary client-side shell scripts in response to hyperlink activations via so-called format/viewer customization options.
- The ability to be controlled by signals to allow use as a full-featured help or information presentation subsystems by existing applications.
- Support for standard World Wide Web authentication scheme, providing security about equivalent to telnet's username/password scheme.
- The availability of customizable encryption hooks to allow external PEM or PGP encryption to be used to request and receive encrypted documents.

Mosaic running on every supported "wired" computer should provide access to the following services:

- Anything served through Gopher (menu-based text retrieval)
- Anything served through WAIS (Wide-Area Information Service)
- Anything served through anonymous FTP (File Transfer Protocol) sites
- Full Archie services (a FTP search service)
- Full Veronica services (a Gopher search service)
- Full X.500 and whois services (Internet phone book services)
- Full finger services (an Internet user lookup program)
- Any library system using PALS (a library database standard)
- Anything on Usenet (mailing lists and news groups)
- Anything accessible through telnet
- Anything in hytelnet (a hypertext interface to telnet)
- Anything in hyper-g (a networked hypertext system in use throughout Europe)
- Anything in the form of UNIX man(ual) pages
- And of course, any HTML-formatted hypertext and hypermedia document

n HTML

The language that Web clients and servers use to communicate with each other is called the *HyperText Transmission Protocol* (HTTP). All Web clients and servers must be able to speak HTTP in order to send and receive hypermedia documents. For this reason, Web servers are often called *HTTP servers*.

The phrase "World-Wide Web" is often used to refer both to the collective network of servers speaking HTTP as well as to the global body of information available using the protocol.

The standard language the Web uses for creating and recognizing hypermedia documents is the *HyperText Markup Language* (HTML). HTML is loosely related to, but strictly technical not a subset of, the **Standard Generalized Markup Language** (SGML), a widely-used document formatting language. Using such a generic markup language, it is possible to markup a document to capture the overall structure of the document, and the precise meaning of the different document elements, as well as their allowable structural interrelationships. This allows an electronic document to be exchanged and displayed, independently from the computer platform, operating system or software application being used.

Design

HTML is a markup language with fairly generic semantics, appropriate for representing information and electronic documents created by a wide range of applications. It is more broadly defined and therefore more generic than many specific SGML applications, but is still completely device-independent. In the original HTML document type definition (Appendix 1: HTML DTD), the accent is primarily on capturing textual data, on representing the structural interrelationships between this textual data, and on defining and representing hyperlinks between textual documents. When the NCSA Mosaic browser started supporting inline images, the basic HTML markup was extended to allow for representation of image data.

Elements

As in SGML, HTML markup is composed of a set of tags for the different structural elements that define a document and guide its display. An HTML element may include a name, some attributes and some text, hypertext or hyperlinks to other media, and will appear in an HTML document as:

```
<tag_name> text </tag_name>
```

or

```
<tag_name attribute_name=argument> text </tag_name>
```

or just

```
<tag_name>
```

For example:

```
<title> My Useful Document </title>
```

and

```
<pre width=60> A lot of text here. </pre>
```

An HTML document is composed of a single element:

```
<html> . . . </html>
```

that is, in turn, composed of head and body elements:

```
<head> . . . </head>
```

and

```
<body> . . . </body>
```

To allow older (before the present efforts at arriving at a standard, fixed DTD) HTML documents to remain readable, `<html>`, `<head>`, and `<body>` are actually optional within HTML documents.

Elements restricted to the head element are:

```
<isindex>
```

Specify index file.

```
<title> . . . </title>
```

Specify document title.

```
<nextid>
```

Set a variable value. Attribute: variable name.

```
<link>
```

Specify relationships to other documents. Attributes: same as the anchor element below.

```
<base>
```

Specify the name of the file in which the current document is stored. This is useful when link references within the document do not include full pathnames (i.e., are partially qualified).

Elements that can be used in the body of the document:

1) text elements:

```
<p> . . . </p>
```

A paragraph that will be formatted before it is displayed on the screen.

```
<pre> . . . </pre>
```

Identifies text that has already been formatted (preformatted) by some other system and must be displayed as is. Preformatted text may include embedded tags, but not all tag types are permitted.

Attribute: width.

```
<listing> . . . </listing>
```

Example computer listing; embedded tags will be ignored, but embedded tabs will work.

```
<plaintext> . . . </plaintext>
```

Include a section of plain text included from some other source.

```
<blockquote> . . . </blockquote>
```

Include a section of text quoted from some other source.

2) hyperlinks or anchors:

```
<a name="target_anchor_name"> . . . </a>
```

Define a target location in a document.

```
<a href="#anchor_name"> . . . </a>
```

Link to a target location in the same file.

```
<a href="URL"> . . . </a>
```

Link to another file.


```
<a href="URL#target_string"> . . . </a>
```

Link to a target location in another file.

Required attributes: one of name or href. Optional attributes: rel, rev, urn, title, methods.

The structure of a *Universal Resource Locator* (URL) is similar to:

```
resource_type://host.domain:port/pathname
```

where the possible resource types include: file, http, news, gopher, telnet, and wais, and the colon followed by the TCP port number is optional. It points to something somewhere on the Internet.

3) headers:

```
<h1> . . . </h1>
```

Most prominent header.

```
<h2> . . . </h2>
```

Lower level headers.

```
<h3> . . . </h3>
```

```
<h4> . . . </h4>
```

```
<h5> . . . </h5>
```

Least prominent header.

```
<h6> . . . </h6>
```

4) logical styles:

```
<em> . . . </em>
```

Emphasis.

```
<strong> . . . </strong>
```

Stronger emphasis.

```
<code> . . . </code>
```

Display an HTML directive.

```
<samp> . . . </samp>
```

Include sample output.

```
<kbd> . . . </kbd>
```

Display a keyboard key.

```
<var> . . . </var>
```

Define a variable.

```
<dfn> . . . </dfn>
```

Display a definition.

```
<cite> . . . </cite>
```

Display a citation.

5) physical styles

```
<b> . . . </b>
```

Bold font.

```
<i> . . . </i>
```

Italics.

```
<u> . . . </u>
```

Underline.

```
<tt> . . . </tt>
```

Typewriter font.

6) lists:

Definition list/glossary:

```
<dl>
  <dt> First term to be defined </dt>
  <dd> Definition of first term </dd>
  <dt> Next term to be defined </dt>
  <dd> Next definition </dd>
</dl>
```

The attribute `compact` can be used to generate a definition list requiring less space.

Unordered list:

```
<ul>
  <li> First item in the list </li>
  <li> Next item in the list </li>
</ul>
```

Ordered list:

```
<ol>
  <li> First item in the list </li>
  <li> Next item in the list </li>
</ol>
```

Interactive menu:

```
<menu>
  <li> First item in the menu </li>
  <li> Next item </li>
</menu>
```

Directory list of items:

```
<dir>
  <li> First item in the list </li>
  <li> Second item in the list </li>
  <li> Next item in the list </li>
</dir>
```

Items should be less than 20 characters long.

7) entities

```
&keyword;
```

Display a particular character identified by a special keyword. For example the entity `&` specifies the ampersand (&), and the entity `<` specifies the less than (<) character. Note that the semicolon following the keyword is required.

```
&#ascii_equivalent;
```

Use a character literally. Again note that the semicolon following the ASCII numeric value is required.

```
<!-- text -->
```

Place a comment in the HTML source.

```
<address> . . . </address>
```

Present address information.

8) elements currently under development:

```
 . . . </img>
```

Include a graphic image for inline display. Implemented by NCSA Mosaic only.

n HTML+

HTML+ is intended to represent a substantial improvement over the existing HTML format offering nested lists, figures, embedded data in foreign formats for equations etc., tables with support for titles and column headings, change bars, entry forms for querying and updating information sources and for use as fill-in questionnaires for e-mailing. HTML+ is designed to be rendered and displayed on an even wider range of different display types and platforms than HTML, supporting plain text, rich formatted text, GIF and JPEG images, MPEG movies and MIME documents. The new HTML+ format has grown out of the experience with HTML, culminating in the desire to add new features, e.g. inline images, tables and fill-out form fields for greater flexibility in querying remote information sources.

Design

The principal design goal behind HTML+ is still to develop a lightweight, practical delivery format for hypertext and hypermedia documents that can be transparently exchanged over the global Internet and can be rendered for display by relatively simple browsers. As a result, HTML+ maintains most of the basic elements of HTML (for backwards compatibility), extending the functionality of these elements where necessary (mostly by the addition of new element attributes), or defines new, more advanced elements (tables, fill-out forms, etc.) to provide additional hypertext and hypermedia functionality.

HTML+ departs slightly from pure presentation independence by allowing authors to specify rendering hints, e.g. to use a bold font for a given type of emphasis. This step was taken to give authors greater control over the final appearance, and is based upon practical experience with the earlier HTML format. In addition, attribute values are used to distinguish different subcategories of markup, rather than adding extra tags. New logical categories of emphasis etc. can be added without the need to change existing browsers. These decisions have made it practical to restrict HTML+ to a very small set of tags.

Elements

We will only discuss the HTML+ elements which are different from the same elements used in HTML, and the HTML+ elements which are completely new to the markup. Note however that the fact that an element is defined in the HTML+ document type definition (Appendix 2: HTML+ DTD) does not mean that it is supported by every WWW browser. Even NCSA Mosaic, arguably the most advanced browser available, supports only a subset of all the new HTML+ elements (at present primarily fill-out forms).

1) changed existing elements:

```
<p> . . . </p>
```

The paragraph element now has additional attributes: `role`, `align`, `indent`.

The emphasis element has been extended to replace the original physical styles elements in HTML.

```
<em> . . . </em>
```

The emphasis element now has additional attributes: `role`, `b` (render in bold font), `i` (render in italic font), `u` (underline text), `tt` (render in typewriter font), `tr` (render in serif -Times Roman- font), `hv` (render in sans serif -Helvetica- font), `sup` (superscript), `sub` (subscript).

```
<pre> . . . </pre>
```

The preformatted paragraph element now has additional attributes: `style`, `tr`, `hv`, `width`.

```
<a> . . . </a>
```

The anchor element now has additional attributes: `name`, `href`, `role`, `effect`, `print`, `title`, `type`, `size`. Most of these attributes are defined for navigational purposes (help avoid reader disorientation).

```
<link> . . . </link>
```

The link element can now be used as a means of describing the relationship between a document and other documents, and has the same attributes as the anchor element. A document can have multiple `link` elements. Typical uses are to indicate authorship, related indexes and glossaries, older or more recent versions etc. Another use is to indicate a stylesheet that contains the author's layout preferences, e.g. for headers and multi-columns displays. Links can also be used to indicate a static tree structure of documents with relationships such as "parent", "next" and "previous". Such implicit links are also useful when one wants to reuse a given subdocument in another independent book, and for referring to non-HTML+ formats such as scanned page images.

```
<img> . . . </img>
```

The image element now has the additional attributes: `src`, `align`, `seethru`, `ismap`. These allow an image to be used as a transparent anchor, or parts of an image to be used as distinct anchors.

2) newly defined elements:

```
<panel> . . . </panel>
```

Floating panel which can be moved around relative to the normal text flow. Should be rendered with a different background and possibly framed. The panel can be anchored to a named point in the document as specified by the `AT` attribute. The panel may be placed at that point or after, but not before.

```
<tbl> . . . </tbl>
```

Tables with titles and column headers. This allows one to define a caption and to differentiate header and data cells. Cells may contain, text, multiple paragraphs, lists and headers. Adjacent cells can be merged, e.g. to define a header which spans two columns. Attributes: `compact`, `border`.

```
<th> . . . </th>
```

A table header cell with attributes `colspan`, `rowspan` and `align`.

```
<td> . . . </td>
```

A table data cell with attributes `colspan`, `rowspan` and `align`.

```
<form> . . . </form>
```

Fill-out form composed from input fields and selection menus. Each `form` should include one or more input elements which can be layed out with normal and preformatted text, lists and tables.

```
<input> . . . </input>
```

Input fields to be used inside a fill-out form. Attributes: `name`, `type` (`text`, `url`, `int`, `float`, `date`, `checkbox`, `radio`), `size`, `value`, `checked`, `disabled`, `error`. Users can alter the value of the input field(s) by typing or clicking with the mouse.

```
<embed> . . . </embed>
```

For embedding information in a foreign format into the HTML+ document. This is very convenient for mathematical equations and simple drawings. The goal here is to eventually be able to publish articles and whole scientific journals on the Web. However, widespread support for formulae is likely to be delayed until most platforms support the relevant symbols fonts (or Unicode).

```
<fig> . . . </fig>
```

Figures elements are defined for a more flexible and versatile display of images than using `img` elements. They can be used to show images and graphics specified in any external format. Attributes: `align`, `cap` (`caption at left`, `right`, `top`, `bottom`), `noflow`, `ismap`, `src`. In the near future, HTML+ may be extended to support simple drawings with embedded hypertext links. One idea would be a line drawing primitive using the `shape` attribute. A better approach is to extend existing drawing formats such as the ANSI Computer Graphics Metafile format (CGM) or Adobe's PDF to include URL-based hypertext links. This extended format could then be used for figures within HTML+ documents.

```
<figa> . . . </figa>
```

A distinct area in the figure to be considered as a separate element.

```
<figt> . . . </figt>
```

Text on top of a figure background, or in a colored background box.

```
<figd> . . . </figd>
```

Textual figure description for display on non-graphical browsers.

```
<changed> . . . </changed>
```

To delineate sections in the document which have changed since a previous version. Change bars are shown for parts of the document which have been changed or amended. Color enhancements may be used to further distinguish the changes, e.g. red lines for strike-through. This mechanism is not intended for representing revision histories, which are better served by traditional change control mechanisms.

n Glossary

This glossary does not claim to be a collection of every term ever used in relation to the World-Wide Web. Due to the abundance of new developments in this young, rapidly growing service on the Internet it is next to impossible to keep track of all the new possibilities which have been or are being introduced. Therefore, this glossary is intended to define a limited but comprehensive set of World-Wide Web specific terms that the reader of this awareness report needs to know in order to understand the possibilities of the Web, and in order to be able to discuss intelligently about the Web with Web insiders.

A

Archie An Internet network service that searches FTP sites for files.

B

browser Software that provides a (GUI) interface to the World-Wide Web.

C

Cello A MS-Windows browser developed at Cornell University, Law School.

CERN The European collective of high-energy physics researchers (European Center for Nuclear Research).

client A computer/program which requests a service of another computer/program.

client-server model An architecture in which systems use and provide distributed services.

D

Doug Engelbart The inventor of many common devices and ideas used in computing today, including the mouse.

F

finger A service that responds to queries and retrieves user information remotely.

FTP File Transfer Protocol. A common method of file transfer across the Internet.

G

Gopher A versatile menu-driven information service.

H

HTML+ The latest version of HTML.

hyper-g A distributed hypertext system mostly popular in Europe.

hyperlinks Connections between hypermedia or hypertext documents and other media.

hypermedia Hypertext that includes or links to other forms of media.

hypertext Text that, when selected, has the ability to jump to another, connected text.

HTML HyperText Markup Language. The standard language used for creating hypermedia documents within the World-Wide Web.

HTTP HyperText Transmission Protocol. The standard language that World-Wide Web clients and servers use to communicate.

hytelnet A hypertext interface to telnet.

I

Internet The global collective of computer networks.

M

man The UNIX manual pages.

Mosaic A popular X Motif/MS-Windows/Macintosh browser developed by the NCSA.

N

NCSA National Center for Supercomputing Applications. A federally-funded organization whose mission is to develop and research high-technology resources for the scientific community.

NNTP News Network Transfer Protocol. A common method by which articles over Usenet are transferred.

P

PALS A standard library database interface.

PGP Pretty Good Protection. Encryption program for e-mail messages.

S

server A computer/program which provides a service to another computer/program.

SGML Standard Generalized Markup Language. A generic markup language for capturing and representing the content and structure of electronic documents.

T

Ted Nelson The inventor of many common ideas related to hypertext, including the word "hypertext" itself.

telnet A program which allows users to remotely use computers across networks.

Tim Berners-Lee The inventor of the World-Wide Web.

U

URL Uniform Resource Locator. Standardized formatted entity within HTML documents which specifies a network service or document to link to.

Usenet The global news-reading network.

V

Vannevar Bush Originator of the concept of hypertext.

Veronica A network service that allows users to search Gopher systems for documents.

W

WAIS Wide-Area Information Service. A service which allows users to intelligently search for information among databases distributed throughout the Internet.

whois A name lookup service.

World-Wide Web The initiative to create a universal, hypermedia-based method of access to information. Also increasingly used to refer to the Internet itself.

X

X.500 A standard which defines electronic mail directory services.

n Appendix 1: HTML DTD

NOTE: No definitive, fixed, standard HTML DTD exists as yet. Some active Web builders even define "standard HTML" as "whatever is supported by the present (X Motif) version of Mosaic". The DTD below is just one of the most recent versions of the many versions available on the Web.

```
<!DOCTYPE HTML [
<!-- Jul 1 93 -->
<!-- DTD definitions -->
<!ENTITY % heading "H1|H2|H3|H4|H5|H6" >
<!ENTITY % list " UL | OL | DIR | MENU ">
<!ENTITY % literal " XMP | LISTING ">
<!ENTITY % headelement " TITLE | NEXTID | ISINDEX" >
<!ENTITY % bodyelement "P | HR | %heading; | %list; |
                        DL | ADDRESS | PRE | BLOCKQUOTE | %literal;">
<!ENTITY % oldstyle "%headelement; | %bodyelement; | #PCDATA">
<!ENTITY % URL "CDATA"
-- The term URL means a CDATA attribute
   whose value is a Uniform Resource Locator,
   as defined. (A URN may also be usable here when defined.) -->
<!ENTITY % linkattributes "NAME NMTOKEN #IMPLIED
                           HREF %URL; #IMPLIED
                           REL CDATA #IMPLIED
                           -- forward relationship type --
                           REV CDATA #IMPLIED
                           -- reversed relationship type to referent data:
                           PARENT CHILD, SIBLING, NEXT, TOP,
                           DEFINITION, UPDATE, ORIGINAL, etc. --
                           URN CDATA #IMPLIED
                           -- universal resource number --
                           TITLE CDATA #IMPLIED
                           -- advisory only --
                           METHODS NAMES #IMPLIED
                           -- supported public methods of the object:
                           TEXTSEARCH, GET, HEAD, etc. --
                           ">
<!-- Document Element -->
<!ELEMENT HTML O O (( HEAD | BODY | %oldstyle; )*, PLAINTEXT?)>
<!ELEMENT HEAD - - ( TITLE? & ISINDEX? & NEXTID? & LINK* & BASE?)>
<!ELEMENT TITLE - - RCDATA
-- The TITLE element is not considered part of the flow of text.
   It should be displayed, for example as the page header or
   window title. -->
```



```

<!ELEMENT ISINDEX - O EMPTY
  -- WWW clients should offer the option to perform a search on
     documents containing ISINDEX. -->

<!ELEMENT NEXTID - O EMPTY>
<!ATTLIST NEXTID N NAME #REQUIRED
  -- The number should be a name suitable for use
     for the ID of a new element. When used, the value
     has its numeric part incremented. E.g. Z67 becomes Z68
  -->

<!ELEMENT LINK - O EMPTY>
<!ATTLIST LINK %linkattributes;>

<!ELEMENT BASE - O EMPTY -- Reference context for URLs -->
<!ATTLIST BASE HREF %URL; #IMPLIED>

<!ENTITY % inline "EM | TT | STRONG | B | I | U |
  CODE | SAMP | KBD | KEY | VAR | DFN | CITE ">

<!ELEMENT (%inline;) - - (#PCDATA)>

<!ENTITY % text "#PCDATA | IMG | %inline;">

<!ENTITY % htext "A | %text;" -- Plus links, no structure -->

<!ENTITY % stext "P | HR | %list; | DL | ADDRESS |
  PRE | BLOCKQUOTE | %literal; | %htext;"
  -- as htext but also nested structure -->

<!ELEMENT BODY - - (%bodyelement; | %htext;)*>

<!ELEMENT A - - (%text;)>
<!ATTLIST A %linkattributes; >

<!ELEMENT IMG - O EMPTY -- Embedded image -->
<!ATTLIST IMG SRC %URL; #IMPLIED -- URL of document to embed -->

<!ELEMENT P - O EMPTY -- separates paragraphs -->
<!ELEMENT HR - O EMPTY -- horizontal rule -->

<!ELEMENT (%heading;) - - (%htext;)+>

<!ELEMENT DL - - (DT | DD | %stext;)*>
<!ELEMENT DT - O EMPTY>
<!ELEMENT DD - O EMPTY>

<!ELEMENT (UL|OL) - - (%htext;|LI|P)+>

<!ELEMENT (DIR|MENU) - - (%htext;|LI)+>

<!ATTLIST (%list;) COMPACT NAME #IMPLIED
  -- COMPACT, etc. -->

<!ELEMENT LI - O EMPTY>

<!ELEMENT BLOCKQUOTE - - (%htext;|P)+ -- for quoting some other source -->

```

```

<!ELEMENT ADDRESS - - (%htext;|P)+>

<!ELEMENT PRE - - (#PCDATA | %inline; | A | P)+>
<!ATTLIST PRE WIDTH NUMBER #IMPLIED>

<!-- Mnemonic character entities. -->
<!ENTITY Aelig "&#198;" -- capital AE diphthong (ligature) -->
<!ENTITY Aacute "&#193;" -- capital A, acute accent -->
<!ENTITY Acirc "&#194;" -- capital A, circumflex accent -->
<!ENTITY Agrave "&#192;" -- capital A, grave accent -->
<!ENTITY Aring "&#197;" -- capital A, ring -->
<!ENTITY Atilde "&#195;" -- capital A, tilde -->
<!ENTITY Auml "&#196;" -- capital A, dieresis or umlaut mark -->
<!ENTITY Ccedil "&#199;" -- capital C, cedilla -->
<!ENTITY ETH "&#208;" -- capital Eth, Icelandic -->
<!ENTITY Eacute "&#201;" -- capital E, acute accent -->
<!ENTITY Ecirc "&#202;" -- capital E, circumflex accent -->
<!ENTITY Egrave "&#200;" -- capital E, grave accent -->
<!ENTITY Euml "&#203;" -- capital E, dieresis or umlaut mark -->
<!ENTITY Iacute "&#205;" -- capital I, acute accent -->
<!ENTITY Icirc "&#206;" -- capital I, circumflex accent -->
<!ENTITY Igrave "&#204;" -- capital I, grave accent -->
<!ENTITY Iuml "&#207;" -- capital I, dieresis or umlaut mark -->
<!ENTITY Ntilde "&#209;" -- capital N, tilde -->
<!ENTITY Oacute "&#211;" -- capital O, acute accent -->
<!ENTITY Ocirc "&#212;" -- capital O, circumflex accent -->
<!ENTITY Ograve "&#210;" -- capital O, grave accent -->
<!ENTITY Oslash "&#216;" -- capital O, slash -->
<!ENTITY Otilde "&#213;" -- capital O, tilde -->
<!ENTITY Ouml "&#214;" -- capital O, dieresis or umlaut mark -->
<!ENTITY THORN "&#222;" -- capital THORN, Icelandic -->
<!ENTITY Uacute "&#218;" -- capital U, acute accent -->
<!ENTITY Ucirc "&#219;" -- capital U, circumflex accent -->
<!ENTITY Ugrave "&#217;" -- capital U, grave accent -->
<!ENTITY Uuml "&#220;" -- capital U, dieresis or umlaut mark -->
<!ENTITY Yacute "&#221;" -- capital Y, acute accent -->
<!ENTITY aacute "&#225;" -- small a, acute accent -->
<!ENTITY acirc "&#226;" -- small a, circumflex accent -->
<!ENTITY aelig "&#230;" -- small ae diphthong (ligature) -->
<!ENTITY agrave "&#224;" -- small a, grave accent -->
<!ENTITY amp "&#38;" -- ampersand -->
<!ENTITY aring "&#229;" -- small a, ring -->
<!ENTITY atilde "&#227;" -- small a, tilde -->
<!ENTITY auml "&#228;" -- small a, dieresis or umlaut mark -->
<!ENTITY ccedil "&#231;" -- small c, cedilla -->
<!ENTITY eacute "&#233;" -- small e, acute accent -->
<!ENTITY ecirc "&#234;" -- small e, circumflex accent -->
<!ENTITY egrave "&#232;" -- small e, grave accent -->
<!ENTITY eth "&#240;" -- small eth, Icelandic -->
<!ENTITY euml "&#235;" -- small e, dieresis or umlaut mark -->
<!ENTITY gt "&#62;" -- greater than -->
<!ENTITY iacute "&#237;" -- small i, acute accent -->
<!ENTITY icirc "&#238;" -- small i, circumflex accent -->
<!ENTITY igrave "&#236;" -- small i, grave accent -->
<!ENTITY iuml "&#239;" -- small i, dieresis or umlaut mark -->
<!ENTITY lt "&#60;" -- less than -->
<!ENTITY nbsp "&#32;" -- should be NON_BREAKING space -->

```

```
<!ENTITY ntilde "&#241;" -- small n, tilde -->
<!ENTITY oacute "&#243;" -- small o, acute accent -->
<!ENTITY ocirc "&#244;" -- small o, circumflex accent -->
<!ENTITY ograve "&#242;" -- small o, grave accent -->
<!ENTITY oslash "&#248;" -- small o, slash -->
<!ENTITY otilde "&#245;" -- small o, tilde -->
<!ENTITY ouml "&#246;" -- small o, dieresis or umlaut mark -->
<!ENTITY szlig "&#223;" -- small sharp s, German (sz ligature) -->
<!ENTITY thorn "&#254;" -- small thorn, Icelandic -->
<!ENTITY uacute "&#250;" -- small u, acute accent -->
<!ENTITY ucirc "&#251;" -- small u, circumflex accent -->
<!ENTITY ugrave "&#249;" -- small u, grave accent -->
<!ENTITY uuml "&#252;" -- small u, dieresis or umlaut mark -->
<!ENTITY yacute "&#253;" -- small y, acute accent -->
<!ENTITY yuml "&#255;" -- small y, dieresis or umlaut mark -->

<!-- deprecated elements -->

<!ELEMENT (%literal;) - - CDATA>

<!ELEMENT PLAINTEXT - O EMPTY>

<!-- Local Variables: -->
<!-- mode: sgml -->
<!-- compile-command: "sgmls -s -p " -->
<!-- end: -->
]>
```

n Appendix 2: HTML+ DTD

NOTE: As with the HTML DTD, no definitive, fixed, standard HTML+ DTD exists as yet. The DTD itself is still "under ongoing construction", although some of its more advanced functionalities (e.g. fill-out forms) are already being supported by the latest version of Mosaic (version 2.4). The DTD below is just one of the most recent versions of the many versions available on the Web.

```
<!DOCTYPE HTMLPLUS [
<!-- DTD for HTML+ It assumes the default <!SGML> declaration.

Markup minimisation should be avoided with the exception of </>
for the endtag. Browsers should be forgiving of markup errors.

Common Attributes:

    id      The id attribute allows authors to name elements such as
            headers and paragraphs as potential destinations for links.
            Note links don't specify points, but rather extended objects.

    index   Allows authors to specify how given headers etc should be
            indexed as primary or secondary keys, where "/" separates
            primary from secondary keys ";" separates multiple entries
-->

<!-- ENTITY DECLARATIONS with <!ENTITY>

<!ENTITY % foo "X | Y | Z"> is a macro definition for parameters and
in subsequent statements, the string "%foo;" is expanded to "X | Y | Z"

Various classes of SGML text types:

    #CDATA      text which doesn't include markup or entity references

    #RCDATA     text with entity references but no markup

    #PCDATA     text occurring in a context in which markup and entity
                references may occur.
-->

<!ENTITY % URL "CDATA" -- a URL or URN designating a hypertext node -->
<!ENTITY % text "#PCDATA|A|IMG|EM|EMBED|INPUT|SP|BR|CHANGED">
<!ENTITY % paras "P|PRE|FIG">
<!ENTITY % lists "UL|OL|DL">
<!ENTITY % misc "TBL|FORM|PANEL|GROUP">
<!ENTITY % heading "H1|H2|H3|H4|H5|H6">
<!ENTITY % table "%text;|P|%heading;|%lists; ">
<!ENTITY % main "%heading;|%misc;|%lists;|%paras;|%text; ">
<!ENTITY % setup "(TITLE? & HTML? & ISINDEX? & NEXTID? & LINK* & BASE?)">

<!--
    <!ELEMENT tagname - - CONTENT> elements needing closing tags
    <!ELEMENT tagname - O CONTENT> elements without closing tags
    <!ELEMENT tagname - O EMPTY> elements without content or closing tags
```

The content definition is:

- a) an entity reference as defined above
- b) a tagname
- c) (brackets enclosing the above)

These may be combined with the operators:

A* A occurs zero or more times
 A+ A occurs one or more times
 A B implies either A or B
 A? A may occur zero or one times
 A,B implies first A then B

-->

```
<!ELEMENT HTMLPLUS O O ((HEAD, BODY) | ((%setup;), (%main;)*))>
```

```
<!ELEMENT HEAD - - (%setup;)>
```

```
<!ELEMENT BODY - - (%main;)*>
```

```
<!-- Document title -->
```

```
<!ELEMENT TITLE - - (#PCDATA | EM)+>
```

```
<!ATTLIST TITLE
```

```
  id      ID      #IMPLIED -- link destination --
```

```
  index   CDATA   #IMPLIED -- entries for index compilation -->
```

```
<!-- Document/Node role for cataloging documents held by servers -->
```

```
<!ELEMENT HTML - O (EMPTY)>
```

```
<!ATTLIST HTML role CDATA #IMPLIED -- home page, index, ... -->
```

```
<!-- Floating panel which can be moved around relative to the normal text
flow. Often rendered with a different background and possibly framed. The
panel can be anchored to a named point in the document as specified by the
AT attribute. The panel may be placed at that point or after, but not
before.
-->
```

-->

```
<!ELEMENT PANEL - - (TITLE?, (%main;)*)>
```

```
<!ATTLIST PANEL
```

```
  id      ID      #IMPLIED -- defines link destination --
```

```
  at      IDREF   #IMPLIED -- anchor point --
```

```
  index   CDATA   #IMPLIED -- entries for index compilation -->
```

```
<!-- Document headers -->
```

```
<!ELEMENT (%heading;) - - (#PCDATA | EM)+>
```

```
<!ATTLIST (%heading;)
```

```
  id      ID      #IMPLIED -- defines link destination --
```

```
  index   CDATA   #IMPLIED -- entries for index compilation -->
```

```
<!-- logical emphasis with optional style hints -->
```

```
<!ELEMENT EM - - (%text;)*>
```

```

<!ATTLIST EM
  role      CDATA      #IMPLIED -- semantic category e.g. CITE --
  b         (b)        #IMPLIED -- render in bold font --
  i         (i)        #IMPLIED -- render in italic font --
  u         (u)        #IMPLIED -- underline text --
  tt        (tt)       #IMPLIED -- render in typewriter font --
  tr        (tr)       #IMPLIED -- render in serif (Times Roman) font --
  hv        (hv)       #IMPLIED -- render in sans serif (Helvetica) font --
  sup       (sup)      #IMPLIED -- superscript --
  sub       (sub)      #IMPLIED -- subscript --
  index     CDATA      #IMPLIED -- entries for index compilation -->

<!-- Paragraphs with different roles and optional style hints -->

<!ELEMENT P - O (%text;)+>

<!ATTLIST P
  id        ID          #IMPLIED -- link destination --
  role      CDATA      #IMPLIED -- semantic role --
  align     CDATA      #IMPLIED -- left, center or right --
  indent    (indent)   #IMPLIED -- indented margins --
  index     CDATA      #IMPLIED -- entries for index compilation -->

<!ELEMENT BR - O EMPTY -- line break -->

<!ELEMENT SP - O EMPTY -- unbreakable space -->

<!-- Preformatted text with fixed pitch font, respecting original spacing
and newlines. Authors can also request proportional fonts. Further control
is possible with EM.
-->

<!ELEMENT PRE - - (%text;)+>

<!ATTLIST PRE
  id        ID          #IMPLIED -- link destination --
  style     CDATA      #IMPLIED -- various styles --
  tr        (tr)       #IMPLIED -- serif (Times Roman) font --
  hv        (hv)       #IMPLIED -- sans serif (Helvetica) font --
  width     NUMBER     #IMPLIED -- e.g. 40, 80, 132 --
  index     CDATA      #IMPLIED -- entries for index compilation -->

<!-- Lists which can be nested -->

<!ELEMENT OL - - (LI | UL | OL)+ -- ordered list -->

<!ATTLIST OL
  id        ID          #IMPLIED
  compact   (compact)   #IMPLIED
  index     CDATA      #IMPLIED -- entries for index compilation -->

<!ELEMENT UL - - (LI | UL | OL)+ -- unordered list -->

<!ATTLIST UL
  id        ID          #IMPLIED -- link destination --
  compact   (compact)   #IMPLIED -- reduced interitem spacing --
  narrow    (narrow)    #IMPLIED -- narrow perhaps multi columns --
  index     CDATA      #IMPLIED -- entries for index compilation -->

```

```

<!-- List items for UL and OL lists -->

<!ELEMENT LI - O (P|%text;)+>

<!ATTLIST LI
  id      ID      #IMPLIED
  src     %URL;   #IMPLIED -- icon for use in place of bullet --
  index   CDATA   #IMPLIED -- entries for index compilation -->

<!-- Definition Lists (terms + definitions) -->

<!ELEMENT DL - - (DT,DD)+ -- DT and DD *MUST* be paired -- >

<!ATTLIST DL
  id      ID      #IMPLIED
  compact (compact) #IMPLIED
  index   CDATA   #IMPLIED -- entries for index compilation -->

<!ELEMENT DT - O (%text;)+ -- term text -- >

<!ELEMENT DD - O (P|QUOTE|UL|OL|%text;)+ -- definition text -- >

<!ATTLIST (DT|DD)
  id      ID      #IMPLIED
  index   CDATA   #IMPLIED -- entries for index compilation -->

<!-- Tables with titles and column headers, e.g.

  <tbl border>
    <tt> An Example of a Table
    <th> <th s="2"> average <th> other <tr>
    <th> <th> height <th> weight <th> category <tr>
    <td> males <td> 1.9 <td> .003 <td> yyy <tr>
    <td> females <td> 1.7 <td> .002 <td> xxx
  </tbl>
-->

<!ELEMENT TBL - - (TT?, (TH|TD|TR|TB)*) -- mixed headers and data -->

<!ATTLIST TBL
  id      ID      #IMPLIED
  compact (compact) #IMPLIED -- if present use compact style --
  border  (border) #IMPLIED -- if present draw borders --
  index   CDATA   #IMPLIED -- entries for index compilation -->

<!ELEMENT TT - O (%text;)+ -- table title -->

<!ATTLIST TT top (top) #IMPLIED -- place title above table -->

<!ELEMENT TH - O (%table;)* -- a header cell -->

<!ATTLIST TH
  colspan NUMBER 1 -- columns spanned --
  rowspan NUMBER 1 -- rows spanned --
  align   CDATA  #IMPLIED -- left, center or right -->

<!ELEMENT TD - O (%table;)* -- a data cell -->

```

```
<!ATTLIST TD
  colspan NUMBER    1      -- columns spanned --
  rowspan NUMBER    1      -- rows spanned --
  align   CDATA     #IMPLIED -- left, center or right -->

<!ELEMENT TR - O EMPTY -- row *separator* not terminator -->

<!ELEMENT TB - O EMPTY -- vertical break of 1/2 line spacing -->

<!-- Forms composed from input fields and selection menus
```

These elements define fields which users can type into or select with mouse clicks. The browser should manage the input focus e.g. with the tab/shift tab keys and mouse clicks.

The enter/return key is then taken to mean the use has filled in the form and wants the appropriate action taken:

- send as query/update to WWW server
- email/fax to designated person

The action is specified as a URL, e.g. "<mailto:dsr@hplb.hpl.hp.com>" You can specify additional mail headers with the MH tag:

```
<MH>Subject: Please add me to tennis tournament</MH>
```

Each FORM should include one or more INPUT elements which can be layed out with normal and preformatted text, lists and tables.

```
-->
```

```
<!ELEMENT FORM - - (MH, (%main;))*>
```

```
<!ATTLIST FORM
  id      ID      #IMPLIED
  action  %URL;   #IMPLIED
  index   CDATA   #IMPLIED -- entries for index compilation -->
```

```
<!ELEMENT MH - - CDATA -- one or more RFC 822 header fields -->
```

```
<!ATTLIST MH hidden (hidden) #IMPLIED -- hide mail headers from view -->
```

```
<!-- INPUT elements should be defined within a FORM element.
```

Users can alter the value of the INPUT element by typing or clicking with the mouse. Use radio buttons for selecting one attribute value from a set of alternatives. In this case there will be several INPUT elements with the same name. Attributes which can take multiple values at the same time should be defined with checkboxes: define each allowed value in a separate INPUT element but with the same attribute name. For checkboxes and radio buttons, the value doesn't change, instead the state of the button shown by the presence or absence of the checked attribute in each element.

The size attribute specifies the size of the input field as appropriate to each type. For text this gives the width in characters and height in lines (separated by an "x"). For numbers this gives the maximum precision.

```
-->
```

```
<!ELEMENT INPUT - O EMPTY>
```



```
<!ATTLIST INPUT
  name      CDATA      #IMPLIED -- attribute name (may not be unique) --
  type      CDATA      #IMPLIED -- TEXT,URL,INT,FLOAT,DATE,CHECKBOX,RADIO--
  size      CDATA      #IMPLIED -- e.g."32x4" for multiline text --
  value     CDATA      #IMPLIED -- attribute value (altered by user) --
  checked   (checked)  #IMPLIED -- for check boxes and radio buttons --
  disabled  (disabled) #IMPLIED -- if grayed out --
  error     (error)    #IMPLIED -- if in error -->
```

<!-- Embedded Data

You can embed information in a foreign format into the HTML+ document. This is very convenient for mathematical equations and simple drawings. Images and complex drawings are better specified as linked documents using the FIG or IMG elements.

Arbitrary 8 bit data is allowed but any occurrences of the following chars must be escaped as shown:

```
"&"    by    "&amp;"
"<"    by    "&lt;"
">"    by    "&gt;"
```

The browser can pipe such data thru filters to generate the corresponding pixmap. The data format is specified as a MIME content type, e.g. "text/eqn" -->

```
<!ELEMENT EMBED - - (RCDATA)>
```

```
<!ATTLIST EMBED
  id      ID      #IMPLIED
  type    CDATA   #IMPLIED -- mime content type --
  index   CDATA   #IMPLIED -- entries for index compilation -->
```

<!-- Figures

The image/drawing is specified by a URL or as embedded data for simple drawings. The element's text serves as the caption. Use the emphasis with style = "credits" to record photo credits etc.

FIGD allows dumb terminals etc to show a textual description in place of the actual sound sequence, image, movie or graphic. -->

```
<!ELEMENT FIG - - (EMBED?, FIGD?, (FIGA|FIGT)*, (%text;)*>
```

```
<!ATTLIST FIG
  id      ID      #IMPLIED
  align   CDATA   #IMPLIED -- position: left, right or center --
  cap     CDATA   #IMPLIED -- caption at left, right, top, bottom --
  noflow  (noflow) #IMPLIED -- disables text flow --
  ismap   (ismap)  #IMPLIED -- server can handle mouse clicks/drags --
  src     %URL;   #IMPLIED -- link to image data --
  index   CDATA   #IMPLIED -- entries for index compilation -->
```

```
<!ELEMENT FIGD - - (%table;) -- figure description -->
```

<!-- Figure anchors designate polygonal areas on the figure which can be clicked with the mouse. The default area is the whole of the figure. This

mechanism interprets mouse clicks locally, and browsers can choose to highlight the designated area (or change the mouse sprite) when the mouse is moved over the area.

Note that polygons may be non-convex or even intersect themselves, thereby complicating the definition of what is enclosed by the polygon. Holes are excluded.

-->

<!ELEMENT FIGA - O EMPTY>

<!ATTLIST FIGA

href %URL; #REQUIRED -- link to traverse when clicked --
 area NUMBERS #IMPLIED -- x1,y1,x2,y2,x3,y3,... -->

<!-- FIGT Text on top of an figure background, or in a colored background box which sits arbitrarily on top of an figure background. The text can include headers, lists and tables etc. The width attribute allows you to limit the width of the text box. The height is then determined automatically by the browser.

FIGT can also be used to position a graphic on top of a picture using an IMG element within FIGT. In this case the chromakey attribute may allow parts of the underlying image to show through.

You can make the whole of the box into a hypertext link. This will act as if it is underneath any hypertext links specified by the overlay markup itself.

-->

<!ELEMENT FIGT - - (%main;)>

<!ATTLIST FIGT

at NUMBERS #IMPLIED -- upper left origin for text --
 width NUMBER #IMPLIED -- given as fraction of picture --
 framed (framed) #IMPLIED -- framed with coloured background --
 href %URL; #IMPLIED -- link to traverse when clicked -->

<!-- inline icons/small graphics

The align attribute defines whether the top middle or bottom of the graphic and current text line should be aligned vertically

The SEETHRU attribute is intended as a chromakey to allow a given colour to be designated as "transparent". Pixels with this value should not be painted. The exact format of this attribute's value has yet to be defined.

Use the FIG tag for captioned figures with active areas etc.

-->

<!ELEMENT IMG - O EMPTY>

<!ATTLIST IMG

src %URL; #REQUIRED -- where to get image data --
 align CDATA #IMPLIED -- top, middle or bottom --
 seethru CDATA #IMPLIED -- for transparency --
 ismap (ismap) #IMPLIED -- send mouse clicks/drags to server -->

<!-- Hierarchical groups for books, chapters, sections etc. -->

```

<!ELEMENT GROUP - - ((TITLE|LINK*), (%main;)*)>

<!ATTLIST GROUP
  id      ID      #IMPLIED
  role    CDATA   #IMPLIED -- book, chapter, section etc. --
  inset   (inset) #IMPLIED -- rendering hint: indent margins -->

<!-- change bars defined by a matched pair of CHANGED elements:

      <changed id=z34> changed text <changed idref=z34>

This tag can't act as a container, since changes don't respect
the nesting implied by paragraphs, headers, lists etc.
-->

<!ELEMENT CHANGED - O EMPTY>

<!ATTLIST CHANGED -- one of id and idref is always required --
  id      ID      #IMPLIED -- signals start of changes --
  idref   IDREF   #IMPLIED -- signals end of changes -->

<!-- Hypertext Links from points within document nodes -->

<!ELEMENT A - - (#PCDATA | IMG | EM | EMBED)*>

<!ATTLIST A
  id      ID      #IMPLIED -- as target of link --
  name    CDATA   #IMPLIED -- backwards compatibility --
  href    %URL;   #IMPLIED -- destination node --
  role    CDATA   #IMPLIED -- role of link, e.g. "partof" --
  effect  CDATA   #IMPLIED -- replace/new/overlay --
  print   CDATA   #IMPLIED -- reference/footnote/section --
  title   CDATA   #IMPLIED -- when otherwise unavailable --
  type    CDATA   #IMPLIED -- for presentation cues --
  size    NAMES   #IMPLIED -- for progress cues -->

<!-- Other kinds of relationships between documents -->

<!ELEMENT LINK - O EMPTY>

<!ATTLIST LINK
  href    %URL;   #IMPLIED -- destination node --
  role    CDATA   #IMPLIED -- role played, e.g. "toc" -->

<!-- Original document URL for resolving relative URLs -->

<!ELEMENT BASE - O EMPTY>

<!ATTLIST BASE HREF %URL; #IMPLIED>

<!-- Signifies the document's URL accepts queries -->

<!ELEMENT ISINDEX - O (EMPTY)>

<!ATTLIST ISINDEX href %URL; #IMPLIED -- defaults to document's URL -->

<!-- For use with autonumbering editors - don't reuse ids,
instead, allocate next one starting from this one -->

```

```
<!ELEMENT NEXTID - O (EMPTY)>

<!ATTLIST NEXTID N NAME #REQUIRED>

<!-- Mnemonic character entities. -->

<!ENTITY AElig "&#198;" -- capital AE diphthong (ligature) -->
<!ENTITY Aacute "&#193;" -- capital A, acute accent -->
<!ENTITY Acirc "&#194;" -- capital A, circumflex accent -->
<!ENTITY Agrave "&#192;" -- capital A, grave accent -->
<!ENTITY Aring "&#197;" -- capital A, ring -->
<!ENTITY Atilde "&#195;" -- capital A, tilde -->
<!ENTITY Auml "&#196;" -- capital A, dieresis or umlaut mark -->
<!ENTITY Ccedil "&#199;" -- capital C, cedilla -->
<!ENTITY ETH "&#208;" -- capital Eth, Icelandic -->
<!ENTITY Eacute "&#201;" -- capital E, acute accent -->
<!ENTITY Ecirc "&#202;" -- capital E, circumflex accent -->
<!ENTITY Egrave "&#200;" -- capital E, grave accent -->
<!ENTITY Euml "&#203;" -- capital E, dieresis or umlaut mark -->
<!ENTITY Iacute "&#205;" -- capital I, acute accent -->
<!ENTITY Icirc "&#206;" -- capital I, circumflex accent -->
<!ENTITY Igrave "&#204;" -- capital I, grave accent -->
<!ENTITY Iuml "&#207;" -- capital I, dieresis or umlaut mark -->
<!ENTITY Ntilde "&#209;" -- capital N, tilde -->
<!ENTITY Oacute "&#211;" -- capital O, acute accent -->
<!ENTITY Ocirc "&#212;" -- capital O, circumflex accent -->
<!ENTITY Ograve "&#210;" -- capital O, grave accent -->
<!ENTITY Oslash "&#216;" -- capital O, slash -->
<!ENTITY Otilde "&#213;" -- capital O, tilde -->
<!ENTITY Ouml "&#214;" -- capital O, dieresis or umlaut mark -->
<!ENTITY THORN "&#222;" -- capital THORN, Icelandic -->
<!ENTITY Uacute "&#218;" -- capital U, acute accent -->
<!ENTITY Ucirc "&#219;" -- capital U, circumflex accent -->
<!ENTITY Ugrave "&#217;" -- capital U, grave accent -->
<!ENTITY Uuml "&#220;" -- capital U, dieresis or umlaut mark -->
<!ENTITY Yacute "&#221;" -- capital Y, acute accent -->
<!ENTITY aacute "&#225;" -- small a, acute accent -->
<!ENTITY acirc "&#226;" -- small a, circumflex accent -->
<!ENTITY aelig "&#230;" -- small ae diphthong (ligature) -->
<!ENTITY agrave "&#224;" -- small a, grave accent -->
<!ENTITY amp "&amp;" -- ampersand -->
<!ENTITY aring "&#229;" -- small a, ring -->
<!ENTITY atilde "&#227;" -- small a, tilde -->
<!ENTITY auml "&#228;" -- small a, dieresis or umlaut mark -->
<!ENTITY ccedil "&#231;" -- small c, cedilla -->
<!ENTITY eacute "&#233;" -- small e, acute accent -->
<!ENTITY ecirc "&#234;" -- small e, circumflex accent -->
<!ENTITY egrave "&#232;" -- small e, grave accent -->
<!ENTITY eth "&#240;" -- small eth, Icelandic -->
<!ENTITY euml "&#235;" -- small e, dieresis or umlaut mark -->
<!ENTITY gt "&#62;" -- greater than -->
<!ENTITY iacute "&#237;" -- small i, acute accent -->
<!ENTITY icirc "&#238;" -- small i, circumflex accent -->
<!ENTITY igrave "&#236;" -- small i, grave accent -->
<!ENTITY iuml "&#239;" -- small i, dieresis or umlaut mark -->
<!ENTITY lt "&lt;" -- less than -->
<!ENTITY ntilde "&#241;" -- small n, tilde -->
```

```
<!ENTITY oacute "ó" -- small o, acute accent -->
<!ENTITY ocirc "ô" -- small o, circumflex accent -->
<!ENTITY ograve "ò" -- small o, grave accent -->
<!ENTITY oslash "ø" -- small o, slash -->
<!ENTITY otilde "õ" -- small o, tilde -->
<!ENTITY ouml "ö" -- small o, dieresis or umlaut mark -->
<!ENTITY szlig "ß" -- small sharp s, German (sz ligature) -->
<!ENTITY thorn "þ" -- small thorn, Icelandic -->
<!ENTITY uacute "ú" -- small u, acute accent -->
<!ENTITY ucirc "û" -- small u, circumflex accent -->
<!ENTITY ugrave "ù" -- small u, grave accent -->
<!ENTITY uuml "ü" -- small u, dieresis or umlaut mark -->
<!ENTITY yacute "ý" -- small y, acute accent -->
<!ENTITY yuml "ÿ" -- small y, dieresis or umlaut mark -->

<!-- dash entities -->

<!ENTITY endash "---" -- En dash -->
<!ENTITY emdash "----" -- Em dash -->

<!-- The END -->

]>
```