Advanced features: HTML stage III

The first bit of this part is not really advanced, but it deals with some things that didn't fit anywhere else: block text, mathematics and tables. The other two sections cover the inclusion of external files and images, and the new form−fill feature of HTML+ which is already included in some clients.

Block text

Block text is text that is set off from the surrounding paragraph by being indented or separated by some space. There are two tags to do this:

- `<address>`...`</address>` which is for surrounding people's addresses. In some browsers it may also cause a font change. It does not honor linebreaks, so if you need to split the address up into separate lines, use the `<br>` tag:
  Peter Flynn
  Computer Center
  University College
  Cork, Ireland
- `<blockquote>`...`</blockquote>` is for block quotations. Browsers may indent this or otherwise set it off from the surrounding text. It does not honor linebreaks, so if you need to split the address up into separate lines, use the `<br>` tag:

  Zerbrochen ist das Steuer und es kracht
  Das Schiff an allen Seiten. Berstend reißt
  Der Boden unter meinen Füssen auf!
  Ich fasse Dich mit beiden Armen an!
  So klammert sich der Schiffer endlich noch
  Am Felsen fest, an dem er scheitern sollte!
  – *Johann Wolfgang von Goethe, 'Torquato Tasso'*

  This tag will be renamed to `<quote>` in HTML+

Mathematics and Tables

HTML as it currently stands does not define any tags for math or tabulations, so you can't type them in directly like you can in TeX or LaTeX. HTML+ may contain math and tables definitions, but in any event, the jury is still out on math in SGML.

However, there are some simple fixes for this:

- for math like $\sum_{i=1}^{n} x_i = \int_0^1 f$, use LaTeX and convert the `.dvi` file to HTML using Nikos Drakos' `latex2html`, from Leeds University;
- for tables, prepare the file as formatted text in fixed−width characters and enclose it in the `<listing>`...`</listing>` tags.

Including external files

There are two mechanisms for including other files at a given point in your document. `<img>` is for including graphic images (pictures) and `<inc>` is for text files. `<inc>` is not in the original HTML but is a new extension.

Images 

Graphics can be included at any point with the `<img src="...">:` tag (defined empty, so there's no end–tag involved). You supply the URL (in quotes) of the graphics file, which must be in one of the following formats:

- GIF, a popular portable graphics format
- JPEG, another format with better compression
- PS, Adobe Postscript, a printer–specification language
- [others may be implemented]

For example, the picture above (the main quad at University College Cork) was specified as

```
<img src="http://curia.ucc.ie/ucc/quad2.gif" align="top" alt="">
```

The optional `align="..."` attribute is an addition to HTML and can be either '`top`', '`bottom`' or '`middle`', which determines if the top or the middle of the image is aligned with the preceding text. By default (if you leave out the `align="..."` attribute), it aligns the bottom of the image with the preceding text. In any case, the text is not moved, it's the image position which is moved.

Users with text–only clients will obviously not get the graphics, but the client may display a marker like `[IMAGE]` where the graphic would normally come. To make this more meaningful, another new optional attribute to the `<img...>` tag has been proposed: `alt="..."` which lets you supply alternate identifying replacement text in quotes which non–graphical browsers can display in place of just saying `[IMAGE]`. Setting it to null (`alt=""`) is a way of preventing non–graphical browsers from displaying anything at all in place of the image.

Take care in including images: many users who are on slow lines or congested networks do not appreciate the wait while their client downloads megabytes of pretty but inessential pictures.

If you include the `ISMAP` attribute in a `<img src="...">` reference, it is possible to arrange for some browsers to transmit mouse coordinates to a mapping server (`mapd`) which will take appropriate action depending on whereabouts within the graphic the mouse was located when it was clicked.

Other non–text included files

Still pictures are not the only things you can include in your document. If your readers have MPEG viewers or audio support, you can use the `<img src="...">` mechanism to refer to movie clips and sound files, which brings W3 into the multimedia field.

The warning above about transmission time applies doubled here: movies over the network take a *lot* of bandwidth.

Including text files

Some systems (notably Mosaic and the NCSA server) allow the in–line inclusion of external files. This is not a part of the original HTML but will likely be included in HTML+.

To include such a file, use the `<inc srv "...">` tag (defined empty, so there's no end–tag) giving the name of the file in quotes. The directory and path should be given as from the server–root of the server being referenced. The file will be interpreted as HTML and formatted accordingly: to display it as plain text, enclose the `<inc srv ...>` tag in one of the fixed–width display elements. Here's a local file I included this way:

```
An SGML document type definition (DTD) is like the description
of a grammar for a programming language.  For example, if you
are working with poetry, you might define:
```

```
   an <anthology> is one or more <poems>

   a <poem> is a <title> followed by one or more <stanzas>

   a <stanza> is one or more lines

   a <line> is a character string

   a <title> is a character string

Documents are marked up in accordance with the grammar defined
in the DTD.
```

WARNING: this uses non−standard syntax (there is no '=' sign between srv and the filename argument, so strictly speaking this is not valid SGML. It is also dependent on using the NCSA server. Details of other forms of the tag and how to include the output of server−side commands are in the NCSA's documentation.

Form−filling

The form−fill feature is a proposed enhancement in HTML+ which Mosaic, Lynx and some other browsers have started to support.

A form is made up of text interspersed with a series of input areas which the user can fill in, and finally send the completed form to a destination you (the author) specify. In most cases this destination is a HTTP server, which will run a script or program to parse (check) the input data and then file or process it or send it to a specified email address. Details of how to specify processing are complex and beyond the scope of this document: you should read (for example) the NCSA's documentation.

Here is a brief overview of the structure of a form and how to specify the input areas (fields). If you want to use forms, you need to have access to a server where you can place the script or program which you want to process the data.

Form definition

A form must be entirely defined within <form...>...</form> tags. The attributes for the start−tag are:

- method="POST", the only value so far defined (in the NCSA's server at least, please let me know if you have more information on others)
- action="*url*", which specifies the URL of the server and script which is to do the processing.

For example:

```
<form method="POST" action="http://abc.edu/htbin-post/myscript">
```

Input areas (fields)

An input area is defined with the <input...> tag (defined empty, so there's no end−tag). This tag takes a variety of attributes which define the name of the field, what type of input it is, the maximum length (in the case of text) or a restricted range of values (in the case of radio or checkbox buttons):

- name="..." lets you name the field (in quotes). This name is passed to the server script so that field values can be identified. In the case of repetitive fields like radio buttons, this name must be repeated identically in each tag;
- type="..." specifies the type of field (in quotes). The main types are:
  - "text", used for names, addresses or any textual data;
  - "number", used for numeric input;
  - "radio", for multi−choice questions where only one choice can be activated (this is the 'check one only' type of question);
  - "checkbox", for multi−choice questions where any number of choices can be activated (this is the 'check all that apply' type of question);

    - `"reset"`, to let the user reset all fields to blank or default;
    - `"submit"`, which the user can activate when the form is completed and ready to send to the server.
- `size=`*nnn* is used with text and number fields to specify the maximum size of text allowed. Giving two numbers separated by a comma indicates multi–line text (the first value is the number of lines, the second the maximum length for all lines);
- `value="..."` specifies the value that this input area will take on if activated (use with non–text fields);
- `checked` can follow the `value="..."` attribute of a radio or checkbox field to specify a default. The button is pre–highlighted to show it is already chosen.

This structure is best shown in an example:

```
<form method="POST" action="http://abc.edu/htbin-post/myscript">
<p>Please answer the following questions:
<ol>
<li>Name: <input type="text" size=40 name="Name"></li>
<li>Company/Institution: <input type="text" size=40 name="Company"></li>
<li>How did you hear about our product? (check all that apply):<br>
<input type="checkbox" name="Source" value="Newspaper">Newspaper
<input type="checkbox" name="Source" value="Radio">Radio
<input type="checkbox" name="Source" value="TV">TV
<input type="checkbox" name="Source" value="Person">Friend/colleague</li>
<li>How many would you like to order? <input type="number"
name="Quantity" size=4></li>
<li>Credit card: <input type="radio" name="CC" value="AMEX">AMEX
<input type="radio" name="CC" value="Visa">Visa
<input type="radio" name="CC" value="Mastercard">Mastercard</li>
<li>Number: <input type="number" name="CCNum" size=16>
and expiry date (mm/yy): <input type="text" name="Expiry" size=5></li>
</ol>
<input type=submit value=" Send this form "> <input type=reset
value=" Reset all values and start over ">
</p></form>
```

Please answer the following questions:

1. Name:
2. Company/Institution:
3. How did you hear about our product? (check all that apply):
   Newspaper Radio TV Friend/colleague
4. How many would you like to order?
5. Credit card: AMEX Visa Mastercard
6. Number: and expiry date (mm/yy):

There's more information on the NCSA's own server.

And now . . .

That's it, there is no more to HTML (yet...wait for HTML+)

Please let me know if you find errors or other deficiencies in this document:

Peter Flynn
Computer Center
University College
Cork, Ireland
`pflynn@curia.ucc.ie`