

Re: Toward WYSIWYG Web Page Authoring

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2005-10/msg00063.html>

- *From:* "Kevin Spencer" <kevin@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Sun, 2 Oct 2005 07:45:41 -0400
-

- > What is the likelihood (and what would it take) of having any true WYSIWYG
- > Web page development capability on the Web – ever?

First, let me give a brief history lesson, followed by some current events, and an optimistic theory about the future. After all, it really isn't logical to talk about the current issue without putting it into historical context, and it isn't possible to solve a problem without knowing something about what brought it about.

All of this is due to a chain of events that began with the invention of HTML and the web browser, less than 2 decades ago. At that time, web pages were just plain text documents. HTML was markup language derived from SGML, and a fairly straightforward markup language for doing some fairly simple formatting of said documents.

HTML had some serious flaws in it, most importantly the use of in-line attributes that were specifically defined for each type of tag, which made it unextensible and inflexible. HTML was originally designed to not be too strict about things like case, closing of tags, placement of tags, etc. My guess is that this was due to the fact that in the beginning, there were no GUIs for developing HTML documents, only text editors. This meant that human error was likely to occur frequently, and the lack of strictness accomodated humand pretty well. No doubt there's more to it, but this is supposed to be "brief."

Despite its flaws, HTML became the standard, by means of the Mozilla browser being the only web browser in existence at the time. However, it wasn't long before other browsers began to appear. This posed a new problem, due to the open nature of the Internet, and the lack of a central authoritative agency to determine standards. New browsers introduced new proprietary tags, and new ways of interpreting HTML, and competition led to the infamous "browser wars," most notably typified in the struggle for supremacy between Microsoft and Netscape, which didn't really resolve any issues about web browsers or HTML.

As the popularity of the WWW increased, HTML was upgraded, and its capabilities were expanded. All of this happened within the context of HTML being seriously flawed.

Re: Toward WYSIWYG Web Page Authoring

It wasn't long before the flaws in HTML began to cause some serious problems. As HTML expanded, it became more complex. New tags and attributes were added, using the same not too strict model, and of course there were even more flawed HTML documents in proliferation on the WWW. HTML documents were becoming unweildy, cryptic, and more and more difficult to parse correctly. The complexity of the language, and the demands of the market brought about the development of various HTML editing software, with a GUI for developing HTML, advertised (somewhat deceptively) as WYSIWYG.

Of course, even at the time of the first GUIs for developing HTML, WYSIWYG was already an impossible goal. New versions of HTML were appearing every several years. New browsers were appearing every several years. Existing browsers were being upgraded every several years. And finally, because of the extant limitations of HTML, combined with market demand for more and more functionality, browser manufacturers were adding proprietary markup to their browsers, and, by association, more "flavors" of HTML were emerging every several years.

The first serious attempt at a rescue, and currently the most popular solution, was the advent of CSS (Cascading Style Sheets). CSS enabled style-centric and functionality-centric markup to be removed from HTML elements themselves, and placed into a separate style sheet. This meant that a web document had the capability of being as extensible as possible, adapting to new enhancements to the language, without revising the HTML itself. Only the style sheet needed to be changed. This was a very hopeful move towards a solution. However, it was not extensible enough, nor did it provide the most optimal separation of logic, content, and style. And of course, there were and are still legacy HTML documents out there all over the place. So, CSS, while providing a bit more sanity to the mix, is still a temporary and incomplete fix.

Browser manufacturers have been somewhat slow to catch up, not notably Internet Explorer of late. However, IE's lateness can be accounted for by its release cycle, which has been seriously delayed since IE6. IE 7 is slated to come out soon, though, and things ought to become more or less equal, and still present problems. After all, how much of the WWW is using CSS? The only possible answer is, more every day.

Still, CSS is a stop-gap measure, a good one nonetheless, but not the best. The best answer has emerged in the form of XML. XML (Extensible Markup Language) is just what it sounds like: an infinitely extensible, infinitely applicable markup language, which can be used not only for web page authoring, but an unlimited host of other purposes, such as data storage, other types of document markup, and other forms of web messaging. In fact, XML is the backbone of SOAP (Simple Object Access Protocol), which enables objects to be accessed across the Internet via HTTP in a cross-platform-compatible format.

XML has been going through quite a few changes as well. XSLT is the XML equivalent of CSS, and presents the most comprehensive and extensible

Re: Toward WYSIWYG Web Page Authoring

solution for the problems that CSS addresses. One of the greatest attractions of XML is its core simplicity. The rules of XML are few. It is, however, strict in format, unlike HTML. This will eventually simplify the browser development process considerably. No longer will browser manufacturers have to make incredibly difficult decisions regarding how to handle poorly-formed markup, whether or not to take which of the document headers seriously, and what to do in this special case, or that one.

XML allows totally granular control over markup, which means that, as the demands for ever-more-sophisticated functionality over the WWW increase, XML will be able to handle these demands without any modification of the standard. It therefore provides stability, in addition to extensibility. It is, to coin a phrase, "the greatest thing since sliced bread."

My theory is that, over the next decade or so, XML will replace HTML on the WWW, and the current issues will disappear (only to be replaced by new and different issues, of course!). This sort of thing has happened before, and will happen again. In the meantime, as always, deal with it. Or, as the young lady says, "I say live it or live WITH it!"

—

HTH,

Kevin Spencer
Microsoft MVP
..Net Developer
Big things are made up of
lots of little things.

"Frankie" <BeansAndTacos@xxxxxxxxxxxx> wrote in message
news:e7tzByxxFHA.2228@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

> Please note that this is NOT a complaint or any sort of rant – but rather
> a serious inquiry about the long-term expectations we can have of the Web
> and Internet as a publication medium. I would appreciate your thoughtful
> feedback on my observations and question (which appears at the end after
> these observations).
>
> The Web as we currently have it is NOT WYSIWYG. I'm specifically referring
> to pages that get rendered by browsers – whether based on HTML, XHTML, or
> XML. In fact, "good Web page design" specifically separates styling from
> content via css (while HTML-specific styling tags like the tag have
> been deprecated). This, by definition, preempts even the *possibility* of
> WYSIWYG Web page designer. What you see in the data/content is ASCII text,
> more or less. What you get in the browser beyond the data/content is
> controlled in large part by the associated css and, separately, browser
> settings (e.g. size of font with which to display everything). Never mind
> that we don't control the size of the final page (monitors come in a
> variety of physical dimensions and resolutions). Also consider that
> Dreamweaver – arguably the most powerful Web page development tool CLEARLY
> states in official documentation that it's not a WYSIWYG editor and that

Re: Toward WYSIWYG Web Page Authoring

- > such a thing is really NOT POSSIBLE on the Web (siting differences in
- > browsers and how each renders pages according to its own interpretation of
- > the standards as the primary cause of that fact).
- >
- > This is all okay for us techies who understand all that and have agreed at
- > least with ourselves and each other to live with it. This is the "state of
- > the medium" and I've heard "if you don't like it, then chose another
- > medium." That's not helpful a helpful statement. An organization may
- > simply not be able to use another medium to accomplish its objectives.
- >
- > While the lack of WYSIWYG on the Web is generally not a problem for us
- > techies, it's definitely a problem for OUR non technical customers and
- > those who don't understand the basic principle of "separate data/content
- > from presentation." (i.e., html/xhtml styled with a separate css page, or
- > CSS-P).
- >
- > This lack of understanding of [the separation of styling from data] is
- > precisely why it's nearly impossible for "non technical" people to create
- > attractive Web pages from scratch. It's also why it's nearly impossible
- > for US TECHIES to create a WYSIWYG Web page editor.
- >
- > As Web developers, everything about this separation of appearance from
- > data/content is JUST FINE as long as WE are in the loop. We know what's
- > going on. But think about the implications of that. This means that in
- > order to get a truly professional-looking Web page, a Web developer MUST
- > be involved at some point. No tool (FrontPage, VS.NET, Dreamweaver, etc)
- > can account for ALL of the relevant factors that go into creating a Web
- > page in a truly WYSIWYG way.
- >
- > THE PUNCH LINE HERE – and a significant problem for all of us (techies and
- > non techies alike) is that we, as Web developers, will never be able to
- > create a tool that will enable NON TECHNICAL users to create
- > professional-looking and behaving Web pages *from scratch*. Period. The
- > non techies are expecting WYSIWYG and are simply NOT CAPABLE of
- > understanding anything other than WYSIWYG. It's just not available on the
- > Web. That's simply a SHOW STOPPER for them.
- >
- > To illustrate the "punch line" described above, think about it from the
- > point of view of someone who is NOT a Web developer and doesn't want to
- > become one. He or she could MUDDLE their way through Word or PowerPoint or
- > PhotoShop or PaintShop Pro or Fireworks Excel and more-or-less create a
- > document he or she is happy with (slide show, jpeg graphic, Word document,
- > charts and graphs, etc). At a minimum they will know what it looks like
- > and what it will look like for everybody else. This same thing can't
- > happen on the Web as we currently know it. The same user who muddles
- > through Word or Fireworks or Excel could NEVER muddle their way through
- > ANY HTML editor and get the equivalent result. They could muddle their way
- > through – but the resulting rendered page would typically be disastrous
- > (from a purely technical perspective) and almost certainly NOT result in
- > what the user wants to create – even on one single browser.
- >

Re: Toward WYSIWYG Web Page Authoring

- > So – my question:
- > What is the likelihood (and what would it take) of having any true WYSIWYG
- > Web page development capability on the Web – ever?
- >
- > –Frankie
- >

- ***Follow-Ups:***

- ◆ ***Re: Toward WYSIWYG Web Page Authoring***
◇ *From: Frankie*

- ***References:***

- ◆ ***OT: Toward WYSIWYG Web Page Authoring***
◇ *From: Frankie*

- Prev by Date: ***Re: dataset question***
- Next by Date: ***Re: tables ;-(***
- Previous by thread: ***Re: Toward WYSIWYG Web Page Authoring***
- Next by thread: ***Re: Toward WYSIWYG Web Page Authoring***
- Index(es):
 - ◆ ***Date***
 - ◆ ***Thread***