TAG ARCHIVES: CHTML

A Brief History of HTML: Part 1

Inspired by one of the tracks at the fantastic EdgeConf London 2015, I realised that the term **Progressive Enhancement** has become confused and ill-defined.

As a reasonably old-school web developer, this is an all too familiar term for me: back in the 1990s JavaScript either didn't exist or was not fully supported, HTML was upgrading from nothing to 2 to 3 to 4, CSS didn't really exist, and there was no such thing as jQuery.

Over the next few articles, I'll be looking back at the evolution of HTML and relevant related technologies and protocols, hoping to arm developers interested in the concept of progressive enhancement with an understanding of the origins of the term and the reason for its existence. This should give you a pause for thought when you need to have a discussion about what lovely

Mobile Data Technologies, Inc.
IPR2024-00246 - Exhibit 2019
Page 1 of 17

shiny new tech to use in your next project.

Progressive Enhancement is not just about turning off JavaScript, not by a long shot. Hopefully soon you'll see what it might encompass over the next few articles.

Part 1: HTML of the 1990s

To really get an appreciation of why progressive enhancement exists, let's take a journey back in time..

It's the 90s. The Spice Girls ruled the Earth before eventually being destroyed by BritPop.

HTML *wasn't even standardised*. Imagine that for a second; different browsers (of which there were very few) implemented some of their own tags. There was no W3C to help out, only a first draft of a possible standard from Tim Berners-Lee whilst he worked this world wide web concept at CERN.

Mobile Data Technologies, Inc.
IPR2024-00246 - Exhibit 2019
Page 2 of 17

Ignore how unstandardised HTML was for a moment; even the underlying protocol for the internet wasn't fully agreed yet.

There was a small-scale Betamax-VHS battle between protocols for suitability of the future of the internet: **HTTP vs Gopher**.

HTTP

We all know what HyperText Transfer Protocol is; clients, servers, sessions, verbs, requests, responses, headers, bodies – oh my!

Hit a server on port 80, execute request using headers and a verb, get a response with headers and body.

But don't forget we're talking about the 90s here; HTTP 1.0 was proposed in 1996, but HTTP 1.1 – the version that is only just being replaced by HTTP2 – was not even standardised until 1999, and has been barely changed for over 15 years.

Gopher

The Gopher protocol is very interesting; it's heavily based on a menu structure for accessing documents intended to feel like navigating a file system.

Hit a server on port 70, get an empty response, send a new line, get a listing back containing titles of documents or directories and magic strings that represent them, client sends back one of these magic strings, server sends that directory listing or document.

This was probably a more suitable match to the requirements of the early web; i.e., groups of documents and listings being

Mobile Data Technologies, Inc.
IPR2024-00246 - Exhibit 2019
Page 3 of 17

requested and searched.

Gopher was released around 1991 but only stuck around for a few years, due to murmurs around licencing which put people off using it for fear of being charged.

Double the fun!

Most browsers of the early 90s supported both protocols. Some of the more "basic" browsers *still do* (GO, LYNX!)

How would you ensure your content is available cross-browser if some browsers support different protocols? Which protocol would you choose? Would you try to implement content on multiple protocols?

Unstandardised HTML

Up until the early-mid 1990s we had browsers which worked on a basic definition of what HTML looked like at the time, and also Gopher.

Browsers at this time generally supported both HTTP and Gopher, up until IE6 and FF 3.6 ish; although there are a selection of browsers that still support Gopher, such as Classica (a fork of Mozilla for Mac), Galeon (GNOME browser based on Gecko), K-Meleon (basically Galeon for Windows), and OmniWeb (for Mac, from the team that bought you OmniGraffle).

There are loads of other browsers, one of my favourites is the text-based browser Lynx, started in 1993 sometime, and is STILL BEING DEVELOPED! This is utterly incredible, as far as I'm

Mobile Data Technologies, Inc.
IPR2024-00246 - Exhibit 2019
Page 4 of 17

concerned. The lynx-dev mailing list has entries from the current month of writing this article and the latest release of Lynx is a little over a year old. Some of the recent messages in the mailing list point out site that refuses to work without JavaScript support. I wonder how it handles angularjs sites?

There was an attempt to standardize HTML with Sir Tim Berners-Lee (one of the inventors of the modern internet – he was not yet Sir at the time though..) putting out a draft of version 1 of the Hypertext Markup Lanauge (yeah, that's what it stands for – had you forgotten?). It didn't really get solidified quickly enough though; the damage was already being done!

HTML standardized

Now we're getting towards the mid 90s; a new draft for standardizing HTML came out and actually got traction; the main browsers of the time mostly implemented HTML 2.0.

During the following few years browsers gradually evolve. HTML 2.0 isn't able to define the sort of world they're now able to create for users, however it's still several years before the next version of HTML has a draft defined, and the browser vendors are getting both restless and innovative.

As such, *they start to support their own elements and attributes*. This is the start of the problem we've had to face for the next couple of decades; by the time HTML 3.2 has been defined, the browsers have left the beaten path to implement strange new functionality. Thought *iFrame* was a standard? Nope; IE only. Firefox instead had the ill-fated *ILayer* and *Layer*.

Mobile Data Technologies, Inc.
IPR2024-00246 - Exhibit 2019
Page 5 of 17

HTML 3.2 had to include *some of the tags* from *some of the browsers* and define them as standard. Obviously, the browser vendors didn't implement all of their competitor's tags whether they were defined in an HTML draft or not. Nor did they stop using their bespoke tags.

We'll pick the impact of this up a bit later.

Mobile development

As we get to the mid-late 90s we can see a deluge of mobile devices starting to enter the market and the types of device vary massively from continent to continent.

These start off as calls and texts only, before supporting some form of more complex data transfer once the networks could support it.

These small(-ish) devices could now access the internet, but they were limited by processing power, memory, storage, screen size, screen capabilities (colour, for example, and lack of fonts), input method, and bandwidth to name but a few. As such, the current HTML 3.2 couldn't be supported, so smart people did clever things.

Two of the main solutions to appear were WML (influenced by HDML Handheld Device Markup Language) and iHTML (which was borne of C-HTML – Compact-HTML).

WML is similar to HTML *but not a subset*, as it contained concepts not required by HTML 3.2 devices; iHTML/C-HTML however, *is* a subset of HTML.

Mobile Data Technologies, Inc.
IPR2024-00246 - Exhibit 2019
Page 6 of 17

However, they both had limited support for more complex elements like tables, jpeg images, and fonts (!)

WML, WAP

Wireless Markup Language and the Wireless Application Protocol were fascinating; many many years ago I created an executive recruitment website's *patented* WAP site with end to end job posting, searching, application, confirmation, etc – all from a WAP device.

WML looks like a mixture of XML and HTML; everything is in a WML document (the *deck*) and the contents of each document are split into one or more *card*s. Each card (or maybe deck) had a limited size, else your content would be truncated by the device's lack of memory or processing power (I remember having to try to intelligently chop content before reaching 1000 chars across cards); this was important when attempting to port content across from an HTML site to a WML site. Only one card is displayed at a time on the device, and navigation between them would be via the device's left and right arrow keys, or similar.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">

<wml>
<card id="intro" title="Intro">
<p>
Hi, here is a quick WML demo
</p>
```


Mobile Data Technologies, Inc.
IPR2024-00246 - Exhibit 2019
Page 7 of 17

```
<p>
   <anchor>
      <prev/> <!-- displays a "Back" link -->
   </anchor>
</p>
</card>

<card id="next" title="Where to we go from here?">
<p>
Now will you take the <a href="#red">red</a> pill or the <a hre
</p>
<p>
   <anchor>
      <prev/>
   </anchor>
</p>
</card>

<card id="blue" title="End of the road">
<p>
Your journey ends here.
</p>
<p>
   <anchor>
      <prev/>
   </anchor>
</p>
</card>

<card id="red" title="Wonderland">
<p>
Down the rabbit hole we go!
```


Mobile Data Technologies, Inc.
IPR2024-00246 - Exhibit 2019
Page 8 of 17

```
</p>
<p>
   <anchor>
      <prev/>
   </anchor>
</p>
</card>
</wml>
```

You can see how it looks like a combination of XML with HTML interspersed. Unfortunately, WAP was sllooowww – you had decks of cards to hide just how slow it was; once you navigated outside of a deck you were pretty much using dial-up to load the next deck.

i-mode

Whilst Europe and the U.S. were working with WML, right at the end of the 1990s Japan's NTT DoCoMo mobile network operator defined a new mobile HTML; a specific flavour of C-HTML

Mobile Data Technologies, Inc.
IPR2024-00246 - Exhibit 2019
Page 9 of 17

(Compact HTML) which was generally referred to as “i-mode” but also responded to the names “i-HTML”, “i-mode-HTML”, “iHTML”, and Jeff.

Ok, maybe not Jeff.

Since this was defined by a network operator, they also had a slightly customised protocol to enable i-mode to work as well as it possibly could on their network.

One of the possible reasons i-mode didn’t make it big outside of Japan could be due to the networks elsewhere in the world simply not being good enough at the time, such that weird gateways had to convert traffic to work over WAP. Yuk; without the concept of *decks* and *cards* for local content navigation, this meant that every click would effectively dial up and request a new page.

i-mode didn’t have the limitations of WML and WAP, and was implemented in such a way that you pressed one button on a handset to access the i-mode home screen and from there could access the official, vetted, commercial i-mode sites without typing in a single `http://`.

Sure, if you really wanted to, you could type in a URL, but a lot of the phones in Japan at the time had barcode readers which means that QR codes were used for i-mode website distribution, and QR codes still remain commonplace thanks to this.

Think about this for a second; the biggest and most powerful mobile network operator in Japan defined a subset of HTML which all of their user’s phones would support, and had a proprietary

Mobile Data Technologies, Inc.
IPR2024-00246 - Exhibit 2019
Page 10 of 17

protocol to ensure it was a snug fit.

All of this was achieved by a team within DoCoMo led by the quite incredible Mari Mastunaga, whom Fortune Magazine selected as one of the most powerful women in business in Japan at the time – think about that for a moment: one of the most powerful women in business, in **I.T.**, in **Japan**, in **1999**. Seriously impressive achievement.

So what?

Let's pause here for now and take stock of the what was happening in terms of web development in the 90s. If you wanted to have a site that worked across multiple devices and multiple browsers, you needed to think about: proprietary elements and attributes outside of HTML 3.2; users with browsers on HTML 2; users on WML; users on i-mode C-HTML; and that's to name but a few concerns.

You needed many versions of many browsers to test on – maybe via VMs, or maybe rely on something like Multiple IE which had its own quirks but allowed you to launch many IE versions at once!

Mobile Data Technologies, Inc.
IPR2024-00246 - Exhibit 2019
Page 11 of 17

But that wouldn't help you test across operating systems; don't forget that IE had a Mac version for a few years. Ouch.

You needed emulators up the wazoo; Nokia had a great WAP one, and if you could fight through the text-heavy Japanese sites you could find some i-mode emulators too.

Progressive Enhancement at the time was less about coding for the lowest common denominator, but using a serious amount of hacky user-agent sniffing to send custom versions of pages to the device; in some cases this would mean reformatting the contents of a page completely via a proxy or similar (e.g., for WML).

You may think this is all way in the past, and if you do unfortunately you're living in a tech bubble; sure, *you* have good wifi, a reliable connection, a decent phone, an up to date laptop with the newest operating system, etc, etc.

Even now i-mode is huge in Japan; if you're not on iPhone, chances are you're on i-mode (or something similar, like Softbank Mobile) and have to think about the current i-mode implementation (yes, they're still making i-mode phones); Facebook recently had issues with their mobile site going awry thanks to how i-mode handles padding on certain cells if you're sending over standard HTML.

If you have an audience in China, remember how many Windows XP IE7 (IE8 if you're lucky) users are there.

Expecting anyone from Burma? Your site better work damn well on

Mobile Data Technologies, Inc.
IPR2024-00246 - Exhibit 2019
Page 12 of 17

Opera Mini as well.

And don't even get me started on Blackberrys.

Next up

We move into the next decade and look at what the 2000s had in store for HTML and the web.

17 July 2015 Robin Osborne

chtml, gopher, http, i-mode, wap, wml, WWW 4 Comments

Search

Search

Recent Posts

Did a Chatbot just improve me?

Burberry X Google Hackathons

Rent vs Buy vs Build

Browser Super Powers: getUserMedia

Archives

- August 2024
- June 2024
- September 2022
- December 2021
- November 2021
- March 2021
- February 2021
- June 2020
- January 2020
- July 2019
- June 2019

Mobile Data Technologies, Inc.
IPR2024-00246 - Exhibit 2019
Page 13 of 17

How to create an Apache-licenced Private WebPageTest setup, and get the Classic Interface

Recent Comments

Matias Autio on A Brief History of HTML: Part 1

Dew Drop – August 16, 2024 (#4246) – Morning Dew by Alvin Ashcraft on Did a Chatbot just improve me?

Robin Osborne on Simple.Data "No ADO Provider Found"

Daniel on Simple.Data "No ADO Provider Found"

Dew Drop – June 5, 2024 (#4200) – Morning Dew by Alvin Ashcraft on Browser Super Powers: getUserMedia

- May 2019
- October 2018
- January 2018
- December 2017
- August 2017
- May 2017
- March 2017
- February 2017
- January 2017
- December 2016
- November 2016
- October 2016
- September 2016
- August 2016
- July 2016
- June 2016
- May 2016
- February 2016
- October 2015
- September 2015
- July 2015
- June 2015
- May 2015
- March 2015
- February 2015
- December 2014
- November 2014
- October 2014
- September 2014
- June 2014
- May 2014

Mobile Data Technologies, Inc.
IPR2024-00246 - Exhibit 2019
Page 14 of 17

- April 2014
- March 2014
- February 2014
- January 2014
- December 2013
- November 2013
- October 2013
- September 2013
- August 2013
- July 2013
- May 2013
- April 2013
- March 2013
- February 2013
- January 2013
- December 2012
- October 2012
- September 2012
- June 2012
- May 2012
- February 2012
- January 2012
- December 2011
- November 2011
- October 2011
- September 2011
- March 2011
- February 2011
- January 2011
- November 2010

Mobile Data Technologies, Inc.
IPR2024-00246 - Exhibit 2019
Page 15 of 17

Categories

- AI
- Asides
- Automation
- AWS
- Azure
- Blog
- Bot
- Build
- Career
- Charity
- Code
- Conference
- EC2
- Environment Automation
- Find of the Day
- intro
- LDNBotFramework
- MVP
- noob
- Performance
- Project
- Rant
- RaspberryPi
- Samsung Smart TV
- Short but sweet
- Team
- Uncategorised

Mobile Data Technologies, Inc.
IPR2024-00246 - Exhibit 2019
Page 16 of 17

- Uncategorized

Proudly powered by WordPress ~ Theme: Syntax by WordPress.com.

Mobile Data Technologies, Inc.
IPR2024-00246 - Exhibit 2019
Page 17 of 17