CS7026: Authoring for Digital Media

The Evolution of Markup Languages
Last week we looked at developing a markup language by writing a DTD using SGML.

If a DTD describes a sufficiently large class of documents it may make sense to construct display software specifically for that class of document and to hardwire the layout rules into it.

That way every document will be laid out consistently by a particular display program.
**Markup Languages**

- This was the approach followed by the first generation of web browsers.

- Most of the formatting of headers, lists, tables and other elements was controlled by the browser, but some details, specifically fonts, type sizes and colours, could be controlled by preferences set by the user.
Alternative option: Provide a separate specification of the layout, parallel to and complementing the DTD.

This layout specification is known as a stylesheet.

For each tag defined in the DTD, a stylesheet provides a rule describing the way in which elements with that tag should be laid out.
Since structural markup can be interpreted on any platform in a logically consistent – if not necessarily visually consistent – manner, it is ideally suited for the markup of documents which are to be transmitted over networks.

When the World Wide Web was being designed SGML was used as the basis of the Hypertext Markup Language (HTML).
HTML

- HTML provides a set of tags suitable for marking up web pages.

- It does not provide any way of defining new tags.

- HTML is defined by an SGML DTD, which describes the structure of the class of documents known as Web pages.

- Like SGML it is concerned solely with identifying the structural elements of a page and not with their appearance.
Originally the WWW was intended as a means of disseminating scientific research, so the DTD for the Web pages contained tags corresponding to the main elements of a scientific paper:

- Title
- Headings that can be nested several levels deep
- Lists of various types
- Some typographical control (bold, italic)
- Tables
- Forms
- Etc.
HTML’s Limitations

- Although adequate for marking up simple mainly textual papers for which they were originally intended, HTML’s layout tags are not sufficient for all the diverse types of material which have found their way onto the WWW.

- Two results:
  - *Browser manufacturers added* *ad hoc* *proprietary extensions to HTML.*
  - *Web Designers used tags as they were never intended.*
XML

- Proposed Solution – provide web designers with a facility to define their own tags.

- SGML can do this, but too unwieldy, slow response times.

- Work on adapting SGML to the Internet led to the definition of a subset, known as XML (eXtensible Markup Language).
**XML**

- Provides all of the important facilities of SGML without the overhead imposed by full SGML.

- XML allows developers to define their own DTDs for any type of document.

- All new browsers now read and interpret (parse) both HTML and XML.
XHTML is an XML-based language that reproduces essentially all the features of HTML, subject to the stricter syntax of XML.

XHTML documents can be processed by both HTML and XML software. So XHTML documents can be read by existing HTML browsers or can be used with software that understands XML.
SGML and XML Family Tree

SGML

HTML

XML

XHTML

MathML

SMIL

SVG
The content of the XHTML 1.0 specification was identical to that of HTML 4.01.

No new elements or attributes were added. The only difference was in the syntax of the language.

Whereas HTML allowed authors plenty of freedom in how they wrote their elements and attributes, XHTML required authors to follow the rules of XML (a stricter markup language) upon which the W3C was basing most of their technologies.
XHMTL 1.0

- The Most Important Differences:
  - XHTML elements must be properly nested
  - XHTML elements must always be closed
  - XHTML elements must be in lowercase
  - XHTML documents must have one root element
**XHTML 1.0**

- Having stricter rules was generally seen to be a good thing. It encouraged a generation of developers to think about valid well-structured code.

- The publication of XHTML 1.0 coincided with the rise of browser support for CSS.

- As web designers embraced the emergence of web standards, led by [The Web Standards Project](https://www.w3.org/standards/about/), the stricter syntax of XHTML was viewed as a “best practice” way of writing markup.
Then the W3C published XHTML 1.1.

While XHTML 1.0 was simply HTML reformulated as XML, XHTML 1.1 was real, honest-to-goodness XML.

That meant it couldn’t be served with a mime-type of text/html.
An aside...

- MIME stands for Multi-purpose Internet Mail Extensions.

- MIME types form a standard way of classifying file types on the Internet.

- Internet programs such as Web servers and browsers all have a list of MIME types, so that they can transfer files of the same type in the same way, no matter what operating system they are working in.

- A MIME type has two parts: a type and a subtype. They are separated by a slash (/). E.g., the MIME type for Microsoft Word files is application and the subtype is msword. Together, the complete MIME type is application/msword.
But if you published a document with an XML mime-type, then the most popular web browser in the world at the time - Internet Explorer - couldn’t render the document.

It seemed as if the W3C, in their quest for standards, were losing touch with the day-to-day reality of publishing on the web.
As far as the W3C was concerned, HTML was finished as of version 4.

They began working on XHTML 2, designed to lead the web to a bright new XML-based future.

Although the name XHTML 2 sounded very similar to XHTML 1, they couldn’t have been more different.
Unlike XHTML 1, XHTML 2 wasn’t going to be backwards compatible with existing web content or even previous versions of HTML.

Instead, it was going to be a pure language, more logical, better-designed, unburdened by the sloppy history of previous specifications.
It was a disaster!
The Schism: WHATWG

- A rebellion formed within the W3C. The consortium seemed to be formulating theoretically pure standards unrelated to the needs of web designers.

- Representatives from Opera, Apple, and Mozilla were unhappy with this direction. They wanted to see more emphasis placed on formats that allowed the creation of web applications.
The Schism: WHATWG

- Things came to a head in a workshop meeting in 2004. Ian Hickson (then from Opera Software), proposed the idea of extending HTML to allow the creation of web applications. The proposal was rejected.

- The disaffected formed their own group: the Web Hypertext Application Technology Working Group, or WHATWG for short.

- Consisted of groups from Opera and Mozilla with Apple cheering by the sideline.
From Web Apps 1.0 to HTML5

- From the start, the WHATWG operated quite differently than the W3C.

- The W3C uses a consensus-based approach: issues are raised, discussed, and voted on.

- At the WHATWG, issues are also raised and discussed, but the final decision on what goes into a specification rests with the editor. The editor is Ian Hickson (who moved to Google).
From Web Apps 1.0 to HTML5

- On the face of it, the W3C process sounds more democratic and fair.

- In practice, politics and internal bickering can bog down progress.

- At the WHATWG, where anyone is free to contribute but the editor has the last word, things move at a faster pace.

- But the editor doesn’t quite have absolute power: an invitation-only steering committee can impeach him.
Initially, the bulk of the work at the WHATWG was split into two specifications: Web Forms 2.0 and Web Apps 1.0.

Both specifications were intended to extend HTML.

Over time, they were merged into a single specification called simply HTML5.
While HTML5 was being developed at the WHATWG, the W3C continued working on XHTML 2.

It wasn’t going well…
In October 2006, Sir Tim Berners-Lee wrote a blog post in which he admitted that the attempt to move the web from HTML to XML just wasn’t working:

“The attempt to get the world to switch to XML, including quotes around attribute values and slashes in empty tags and namespaces, all at once didn’t work.”
Reunification

- A few months later, the W3C issued a new charter for a HTML Working Group.

- Rather than start from scratch, they wisely decided that the work of the WHATWG should be used as the basis for any future version of HTML.
Reunification

- All of this stopping and starting led to a somewhat confusing situation.

- The W3C was simultaneously working on two different, incompatible types of markup: XHTML 2 and HTML 5 (note the space before the letter five).

- Meanwhile a separate organization, the WHATWG, was working on a specification called HTML5 (with no space) that would be used as a basis for one of the W3C specifications!
WHATWG Process

- The WHATWG process has been extraordinarily open – anyone could join the WHATWG mailing list and contribute to the spec.

- Every email was read by Hickson or the core WHATWG team (included Brendan Eich (inventor of JavaScript and Mozilla CTO), David Hyatt (Safari and WebKit Architect) and Hakon Wium Lie (Opera CTO)).

- Good ideas were implemented and bad ideas rejected regardless of the source.

- Good ideas were adopted from Twitter, Blogs, IRC...
XHTML is dead

- The fog of confusion began to clear in 2009. The W3C announced that the charter for XHTML 2 would not be renewed.

- The format had been as good as dead for several years; this announcement was little more than a death certificate.
Meanwhile, authors who had been writing XHTML 1 in order to enforce a stricter writing style became worried that HTML5 would herald a return to sloppy markup.

As you’ll see, that’s not necessarily the case. HTML5 is as sloppy or as strict as you want to make it.

I will want you to be strict!
Not so happy ever after…

For a number of years, both groups then worked together.

In 2011, however, the groups came to the conclusion that they had “different goals”:

“the W3C wanted to publish a "finished" version of "HTML5", while the WHATWG wanted to continue working on a Living Standard for HTML, continuously maintaining the specification rather than freezing it in a state with known problems, and adding new features as needed to evolve the platform” (WHATWG).
Not so happy ever after…

“Since then, the WHATWG has been working on this specification (amongst others), and the W3C has been copying fixes made by the WHATWG into their fork of the document, as well as making other changes, some intentional and some not, with no documentation listing or explaining the differences.”

http://developers.whatwg.org/introduction.html
The Timeline of HTML5

- This left two groups working on HTML5:
  - The WHATWG creating an HTML5 specification using its process of “commit then review.”
  - The W3C HTML Working Group taking that specification and putting it through its process of “review then commit.” As you can imagine, it was an uneasy alliance.

- Still, there seemed to finally be some consensus about that “space or no space?” question (it’s no space).
The Timeline of HTML5

- In September 2014, W3C moved HTML5 to Proposed Recommendation.

- On 28 October 2014, HTML5 was released as a stable W3C Recommendation, meaning the specification process is complete.

- However there wasn’t a single point in time that it was declared that the language was ready to use. Instead, parts of the specification have been used as web browsers started to support those features (around 2010).
The Timeline of HTML5

- Remember, HTML5 isn’t a completely new language created from scratch. It’s an evolutionary rather than revolutionary change in the ongoing story of markup.

- If you are currently creating websites with any version of HTML, you’re already using HTML5.