XML and the Third-Generation Web

Maik Stührenberg
Bielefeld University
maik.stuehrenberg@uni-bielefeld.de
@mstuehrenberg
“The combination of hypertext and a global Internet started a revolution. A new ingredient, XML, is poised to finish the job”

“XML and the Second-Generation Web” by Jon Bosak and Tim Bray, 1999
XML is a success story

- It is everywhere (wait for Steven Pemberton's closing talk)
- It is used for storing large text corpora in the Digital Humanities and complex technical documentation (TEI, Docbook)
- Application of XML-based markup languages vary from annotating music (MusicXML, MEI) to vector graphics (SVG) and geo-spatial information (KML)
- It runs business transactions, orders, and invoices (UBL)
Most of XML's companion standards are successful as well:

- **XSLT** stylesheets are used thousands of times a day for delivering web pages, business reports, publishing books and the like

- **XQuery** has become a serious query language replacing former proprietary languages for XML annotated data in native XML databases

- **RELAX NG and XSD** have gained ground as robust and well-supported document grammar formalism
so... everything's fine? Just sing another Happy Birthday song for XML?
Well, no!

Just think about the setting XML was developed in more than 15 years ago...
Let's use a time machine...
Let's use a time machine...
...a real one!
Structure of the reminder of this talk

The First-Generation Web and XML
The Second-Generation Web
Why XML failed on the Web
The Third-Generation Web
Conclusions
We're in the late 1990s...

- SGML, although successful in publishing, is way too complex to be used on the internet
- Tim Berners-Lee invents HTML as an SGML application
- It's the time of the First-Generation Web
The First-Generation Web

Architecture:

- Focussed on large servers – every change driven by the client has to be processed by the server which sends the result back to the client

Markup:

- HTML mixes structure and layout (<font> anyone?)
- HTML's forms are not expressive enough
- HTML's linking abilities are not 'hyper' enough (no in-place substitution, no third-party-links,...)

- Some smart people postulate a simplification of SGML: XML
XML's design goals


- First principle: “XML shall be straightforwardly usable over the Internet”
- Second principle: “XML shall support a wide variety of applications”
- Fourth principle: “It shall be easy to write programs which process XML documents”
In 1997, Bray et al. publish two working drafts:

- “Extensible Markup Language (XML”) addresses basic questions about the syntax and the formal model
- “Extensible Markup Language (XML): Part 2. Linking” specifies constructs to describe links and pointers in XML instances

→ first steps to the later W3C Recommendations XLink and XPointer
In 1999, Bosak and Bray published this article in the Scientific American.

They postulate the Second-Generation Web with XML being a first-class citizen and enabling technology.
The Second-Generation Web

Ingredients of the 2\textsuperscript{nd} gen Web according to Bosak and Bray:

- Unicode for a really World Wide Web
- Structural and semantic information related to specific application scenarios
- Processing of relevant information by the client → reduced network traffic
- RDF metadata for faster and more accurate retrieval
- XLink for more flexible linking
- XSLT for styling XML output
- Java applets
A brave new world...

To make the 2nd gen Web a reality Browsers should...

- ...ship with an XML parser (for instances and DTDs)
- ...support XLink and XForms
- ...include an XSLT processor
On the way to Second-Generation Web:

- XHTML 1.0 was a Recommendation, XHTML 1.1 made first steps to a modular web markup language
- Microsoft had MSXML used by IE
- Mozilla had a XSLT 1.0 processor and basic XLink support in Firefox
- The XForms extension provided an XForms processor for Firefox 2.x
- X-Smiles was available as a dedicated XML browser
- RSS and Atom were established as XML-based formats
...and then, something's happened
Nobody wanted to use XML on the Web – or at least no one was able to do it right (see Evan Goer's XHTML 100)
Let's be honest:

- One reason why HTML was so successful was the forgiveness of browsers' parsers (browsers' parsers were no SGML parsers at all)
- XML parsing requires draconian error handling which is painful for Joe Average's website – compare this to an American gas pump
- Only few people have been able to use XHTML the correct way (including the application/xhtml+xml MIME type – thanks to IE)
- Java is currently working as a demonstration for Oracle's update service
Dead end for further Web development?

- Browser vendors wanted to extend HTML's capabilities, the W3C wanted to stick with XML (including XML Namespaces for XHTML 2.0)
- In 2004 browser vendors founded the Web Hypertext Application Technology Working Group (WhatWG) and started development for Web Forms 2.0 and other WWW specs, resulting in HTML5
- 2009: XHTML 2.0 working group's charter has been expired
Dead end for XML on the Web!

- Without XML as base technology, XForms, XLink and XSLT are only second-class citizens for browser vendors.
- XML's great success in other fields make it nearly impossible to make changes to basic concepts to create something such as XML 2.0 (see James Clark's blog entry about MicroXML) – think of Norman Walsh's keynote.
The Third-Generation Web

Where are we now?

- W3C will publish the final HTML5 Recommendation in 2014, WhatWG will develop HTML further as a living standard.
- HTML5 uses a slightly different syntax compared to HTML 4.01 and to XHTML – however, there is an XML serialization (XHTML5).
- MathML and SVG are hard-coded into HTML5.
The Third-Generation Web

- HTML5's DOM is still different from XML's DOM (see Norman Walsh's talk at XML Prague 2011 and the panel discussion at XML Prague 2011) – in general one can not expect XML tools to process HTML5)

http://livedom.validator.nu/
The 3rd gen web is mobile:

- Mobile devices such as smartphones have limited processing power to include full XML parser and XSLT processor
- Web apps everywhere – the Web is transforming into an application platform
- HTML5 is much easier to process, especially since it is the first specification that includes detailed instructions for how browsers should handle malformed and legacy markup (based on legacy behavior)
XML's standing

XML's current situation in the browser:

- XSLT support in the browser is still the same as it was 10 years ago
- No support for XSD or RNG validation in the browser
- JSON competes as a lightweight data format

But...
...there is light on the horizon!
XSLT in the browser

- Saxon-CE extends/replaces browser's XSLT engines and is cross-platform thanks to JavaScript
- It changes the XSLT paradigm, allowing for transforming and injecting chunks of information – think of Eric van der Vlist's talk

http://www.masereeuw.nl/nevermind/
And that's just the beginning – there are already various examples around...

XSLT in the browser

http://datenverdrahten.de/xslt2/saxon-ce/studis/
XForms in the browser

- OrbeonForms, XSLTForms, betterForm, etc. provide support for current and upcoming versions of XForms
- Examples can be found in various locations and projects

http://www.agencexml.com/xsltforms/calculator.xml
• Even XQuery in the browser is possible (XQIB)
So, ...

the answer to the question if there is XML on the 3rd gen Web is 'no'?
Well, no!
You won't get XML support in web browsers because of the following deal-brakers (imho):

- The irony of draconian error handling: Error recovery would have brought much more sympathy to the idea of XML in the browser (see Anne van Kesteren's XML5 project at http://code.google.com/p/xml5/ and James Clark's initial blog post about MicroXML)

- XML Namespaces 
  Love it or hate it – they are complicated (although there are options, see Liam Quinn's paper at Balisage 2009)
There is XML in the Third-Generation Web:

- On the server side (common for years)
- On the client side if the community embraces JavaScript as bridging technology – products such as Saxon-CE are great examples
- In addition, XML standards and applications play nicely with JSON- and XML-encoded data (wait for William Candillion's talk)
Conclusions

Comparing the 2nd with the 3rd gen Web

XML → JavaScript → (X)HTML

XML → (X)HTML
Recommendations

What to do?

- HTML5 won't go away – it will be even used for non-web-content such as ebooks (see Sanders Kleinfeld's talk at this year's Balisage)
- Use the XHTML5 syntax or even polyglot HTML5/XHTML5 documents – also have a look at Norman Walsh's talk at XML Prague 2011 and think about Eric's different solutions for XML in hostile environments
Recommendations

What else to do?

- Play a part in the Third-Generation Web standardization process and embrace (and push forward) XML's good ideas (thinking of the Custom Element Spec)
Thank you for your attention!

Questions? Comments?

maik.stuehrenberg@uni-bielefeld.de
@mstuehrenberg