Decentralized Extensibility & HTML 5

(An introduction to the debate)

Noah Mendelsohn
Distinguished Engineer – IBM Corp.
Co-chair: W3C Technical Architecture Group

W3C Technical Plenary
4 November 2009

Disclaimer

The opinions expressed in this presentation are not necessarily those of the TAG or of my employer, IBM.

The Basics

What is Decentralized Extensibility?

The ability for a language to be extended by multiple parties who do not explicitly coordinate with each other.

What sorts of extensions?

- Elements
  - SVG
    ```
    <svg:circle>
    ```
  - MathML, etc.
    ```
    <math:mrow>
    ```
  - FBML
    ```
    <fb:if-is-friends-with-viewer uid="12345">
        Hey you guys are friends!
      <fb:else>
        How come you don't like me?
      </fb:else>
    </fb:if-is-friends-with-viewer>
    ```
- Attributes
  - RDFa
    ```
    <span property="cal:dtstart"
            content=
                   "2007-09-16T16:00:00-05:00"
            datatype="xsd:dateTime">
    ```
- Data values
  - Enumerated attribute values
    ```
    <link rel="…your relation here…">
    ```

Includes changes intended for widespread use, as well as local extensions

Some Pros and Cons of Decentralized Extensibility

Why DE is good – architects' view

- Modularity is good
- Separation of concerns is good
- The Web is too big for any central group to invent or cooordinate all needed extensions to languages like HTML

Q.E.D.

Why DE is good – real world view

- Easier to reuse the pieces
  - SVG can be hosted in many languages, not just HTML
- Copy/paste works across those different container languages
- Shared implementation code
  - The same SVG parser/renderer might be usable in lots of containers
- Modular tooling
  - the same SVG tooling can help you build SVG for many host languages
- Reduced duplication of: user training, documentation, etc.
- Testing: separately developed pieces can be tested separately (up to a point)
- *Allow for experimentation and competition: marketplace decides which are the best enhancements*

Why *not* provide decentralized extensibility?

- Nobody's found a completely painless way to do DE — we'll explore why in a minute
- Controversy: not everyone believes HTML extensions will be needed very often anyway
- *Many of the mechanisms proposed to avoid name collisions are ugly and/or complicated*
- With DE, it can be hard to move experimental extensions into the core
  - `<xxx:table>` → `<table>`

Note: as we'll see, the main controversies involve name collisions

Extensibility in HTML 5

Does HTML 5 provide DE?

Yes!

http://dev.w3.org/html5/spec/Overview.html#other-applicable-specifications:

"When vendor-neutral extensions to this specification are needed, either this specification can be updated accordingly, or an extension specification can be written that overrides the requirements in this specification. When someone applying this specification to their activities decides that they will recognise the requirements of such an extension specification, it becomes an applicable specification for the purposes of conformance requirements in this specification. "

What the text/html serialization of HTML 5 does not *provide are mechanisms like XML Namespaces that help to avoid naming conflicts or help exploit existing vocabularies.*

HTML 5 extension points (partial list)

- New element/attribute markup
- `@class` attribute
- `@rel` attribute on `<a>` and `<link>`
- `<meta name="" content="">`
- `<script type="">` with a custom type to embed raw data
- `<embed>` & `<object>`
- `@data-` attributes (new in HTML5)
- `@itemprop` (HTML 5 microdata)
- And yes, XML Namespaces (*but only using the XML serialization -- not in the more commonly deployed text/html*)

Coordinating extensions to HTML 5

- Assumption is that HTML Working Group will coordinate adoption of *major* new function
- HTML Recommendation will be updated
- New data values, types, etc.
  - IANA registries
    - MIME types
    - Character sets
  - Wikis
    - Name attributes on <meta> (http://wiki.whatwg.org/wiki/MetaExtensions)
    - Pragma extensions (http://wiki.whatwg.org/wiki/PragmaExtensions)
    - <link> rel attributes (http://wiki.whatwg.org/wiki/RelExtensions)

Some issues w/extensibility in HTML 5

- XML not fully supported in text/html media type
- Namespaces not fully supported in text/html media type
- Therefore: existing XML-based capabilities may not integrate, especially with text/html tag soup (which is the common case!)
- The specification has been criticized as including too much *(see next slide)*

HTML builds in what could be extensions

- Graphics:
  - Canvas
  - SVG (status not settled): current HTML 5 spec discusses only element categorization and encourages XML-compatible export

- Data
  - Data-* attributes
  - Microdata (itemprop)
  - RDFa mapping (removed from recent drafts?)

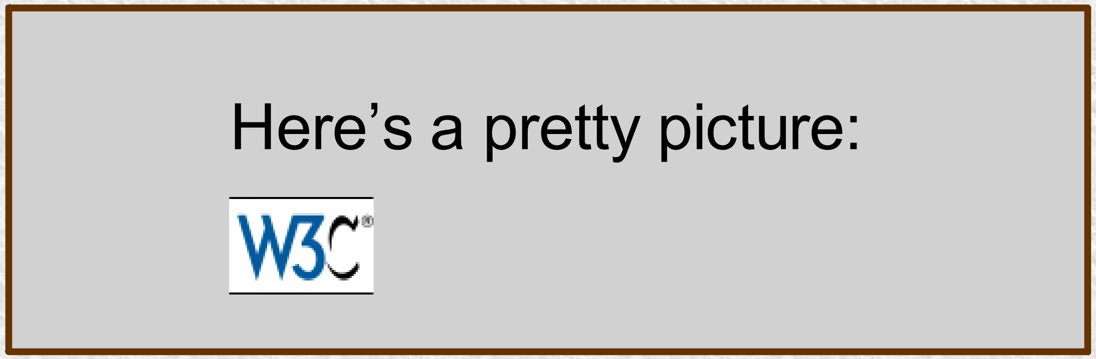
The status of several of the above features is still being resolved.

Name Collisions, Self-describing Markup & XML Namespaces

Pros and cons of namespaces

```
<html>
  <body xmlns:mosai          ttp://ncsa.uiuc.edu/tags”>
    <p>Here’s a pretty picture:</p>
    <mosaic:img
        src=http://www.w3.org/Icons/w3c_home>
  </body>
</html>
```

We know which <img> this is…


Here’s a pretty picture:

Pros and cons of namespaces

```
<html>
  <body xmlns:mosai      ttp://ncsa.uiuc.edu/tags">
    <p>Here's a pretty picture:</p>
    <mosaic:img
        src=http://  w.w3.org/Icons/w3c_home>
  </body>
</html>
```

We know which <img> this is…

Search engines & tools can look for exactly this tag

Here's a pretty picture:

Pros and cons of namespaces

```
<html>
  <body xmlns:mosaic="http://ncsa.uiuc.edu/tags">
    <p>Here's a pretty picture:</p>
    <mosaic:img
        src=http://www.w3.org/Icons/         ome>
  </body>
</html>
```

The namespace is "on the Web"…there's a good chance we can find documentation here

Here's a pretty picture:

W3C

Pros and cons of namespaces

```
<html>
  <body xmlns:mosaic="http://ncsa.uiuc.edu/tags"
        slac="http://slac.stanford.edu/tags">
    <p>Here's a pretty picture:</p>
    <mosaic:img
        src=http://www.w3.org/Icons/w3c_home>
    <other:img
        href="h    //www.w3.org/Icons/w3c_home>
  </body>
</html>
```

Same local name on two tags…
namespaces keep them unique

Pros and cons of namespaces

```
<html>
  <body xmlns:mosaic=“http://ncsa.uiuc.edu/tags”
        slac=“http://slac.stanford.edu/tags”>
    <p>Here’s a pretty picture:</p>
    <mosaic:img
        src=http://www.w3.org/Icons/w3c_home>
    <other:img
        href=“http://www.w3.org/Icons/w3c_home>
  </body>
</html>
```

Same local name on two tags… namespaces delay decisions about who gets to define common shortnames

Pros and cons of namespaces

This markup is truly ugly!!

Users *hate* typing this and can’t get it right.

```
<html>
  <body xmlns:mosaic=“http://ncsa.uiuc.edu/tags”>
    <p>Here’s a pretty picture:</p>
    <mosaic:img
        src=http://www.w3.org/Icons/w3c_home>
  </body>
</html>
```

Here’s a pretty picture:

W3C

Pros and cons of namespaces

URIs are hard to remember…

…tends to be true of all schemes for unique decentralized names!

```
<h
  <body xmlns:mosaic=“http://ncsa.uiuc.edu/tags”>
    <p>Here’s a pretty picture:</p>
    <mosaic:img
          src=http://www.w3.org/Icons/w3c_home>
  </body>
</html>
```

Here’s a pretty picture:

Pros and cons of namespaces

Prefixes add architectural complexity.

```
<html>
  <body xmlns:mosaic="http://ncsa.uiuc.edu/tags">
    <p>Here's a pretty picture:</p>
    <mosaic:img
         src=http://www.w3.org/Icons/w3c_home>
  </body>
</html>
```

Here's a pretty picture:

Pros and <u>cons</u> of namespaces

So do default namespaces.

```
<html>
  <body xmlns=“http://ncsa.uiuc.edu/tags”>
    <p>Here’s a pretty picture:</p>
    <img
         src=http://www.w3.org/Icons/w3c_home>
  </body>
</html>
```

Here’s a pretty picture:

Pros and cons of namespaces

```
<html>
  <body xmlns=“http://ncsa.uiuc.edu/tags”>
    <p>Here’s a pretty picture:</p>
    <img
        src=http://www.w3.org/Icons/w3c_home>
  </body>
</html>
```

Namespaces tend to break source-level copy/paste.

```
<html>
  <body>
    <p>Here’s a pretty picture:</p>
    <img
        src=http://www.w3.org/Icons/w3c_home>
  </body>
</html>
```

Pros and cons of namespaces

```
<html>
  <body xmlns="http://ncsa.uiuc.edu/tags">
    <p>Here's a pretty picture:</p>
    <img
        src=http://www.w3.org/Icons/w3c_home>
  </body>
</html>
```

```
<html>
  <body>
    <p>Here's a pretty picture:</p>
    <img
        src=http://www.w3.org/Icons/w3c_home>
  </body>
</html>
```

Namespaces tend to complicate tools that auto-generate markup (e.g. XQuery serialization).

Pros and cons of namespaces

```
<html>
  <body xmlns="http://ncsa.uiuc.edu/tags">
    <p>Here's a pretty picture:</p>
    <img
       src=http://www.w3.org/Icons/w3c_home>
  </body>
</html>
```

Namespaces tend to break DOM-level updates (e.g. `innerHTML`).

```
<html>
  <body>
    <p>Here's a pretty picture:</p>
    <img
       src=http://www.w3.org/Icons/w3c_home>
  </body>
</html>
```

Pros and cons of namespaces

```
<html>
  <body xmlns:mosai          ://ncsa.uiuc.edu/tags">
    <p>Here's a pr    ety picture:</p>
    <mosaic:img
        src=http://www.w3.org/Icons/w3c_home>
  </body>
</html>
```

If we eventually want this as part of core HTML, *how do we get rid of the prefix?*

Here's a pretty picture:

W3C®

Pros and cons of namespaces

```
<html>
  <body xmlns:mosai          ://ncsa.uiuc.edu/tags">
    <p>Here's a pr    ty picture:</p>
    <mosaic:img
        src=ht    //www.w3.org/Icons/w3c_home>
  </body>
</html>
```

If we eventually want this as part of core HTML, how do we get rid of the prefix?

How do we convince search engines and tools that `<img>` (new content) and `<mosaic:img>` (early adopter content) are the same tag?

Here's a pretty picture:

W3C

Proposals for namespaces in HTML 5

- Liam Quin: “Automatic XML Namespaces” (http://www.w3.org/2009/Talks/08-quin-balisage-namespaces/)
  - Namespace definition files provide prefix bindings
  - Default bindings can be associated with dated version of HTML specification
- Microsoft: “Distributed Extensibility Submission” (http://lists.w3.org/Archives/Public/public-html/2009Sep/att-1216/MicrosoftDistributedExtensibilitySubmission.htm)
  - Mostly supports ordinary `xmlns:pref` syntax
  - Some special handling for HTML, SVG, MathML namespaces
  - Several proposed options for unbound prefix handling
  - Optional: allow default namespace binding *except on root element*
  - Optional: magic namespace assigned for unbound prefixes
  - Optional: predefine prefixes such as `html:`, `math:`, `svg:`
  - Removes Internet Explorer restriction that prefixes be bound on root element

Conclusion

Summary: contentious issues

- There is disagreement as to how often extensions will be needed and whether name collisions would tend to be a problem.
- There is disagreement as to whether central coordination through the HTML Working Group & Wikis is sufficient
- There is disagreement about whether it's practical to provide decentralized mechanisms to avoid name collisions
  - Namespaces are ugly, but widely deployed, usable in XHTML
  - Namespaces can complicate progression of a feature from experimental to core
  - Proposals have been made for adapting namespaces to HTML 5
- There is disagreement as to how much to compromise to maintain compatibility with XML
  - Making namespaces work for text/html documents
  - Consistent application of prefixes in the DOM, etc.
- There is disagreement as to which capabilities should be split out from HTML 5 and which existing Recommendations to make usable in HTML 5:
  - Microdata
  - RDFa mappings
  - SVG
  - Canvas
  - Etc.
- There is disagreement as to whether RDFa in particular is needed

Summary: why it matters

Decisions about decentralized extensibility will affect:

- Whether HTML 5 will adapt well as new capabilities are needed on the Web
- Who will be able to create and deploy enhancements to HTML
- Whether HTML 5 will be convenient to use
- Whether HTML 5 will be compatible with existing Web content
- Whether HTML 5 will have good synergy with XML tools and languages
- Whether HTML 5 itself can evolve, in future versions, into an increasingly robust framework for Web documents

Thank you!