What Are Microformats

March 23, 2005

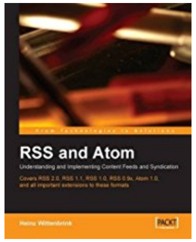
Micah Dubinko

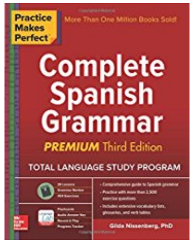
> "The cheapest, fastest and most reliable components of a computer system are those that aren't there." —Gordon Bell

The phrase "XML world" paints an apt word picture, for on top of a bedrock specification lives a thriving ecosystem, inhabited by complex layers of specifications, products, and personalities, both individually and in consortia. Like any ecosystem, XML world is subject to Darwinian natural selection and periodic adjustments. The best ideas tend to stick around.

The idea of *microformats* is particularly being explored of late. Previously, XML-Deviant discussed several microformats in the context of Google's good example of utilizing new technologies. But what exactly is a microformat? A primary source for microformat information is Technorati's developer wiki entry, which doesn't define the term, but rather illustrates around it. In summary, the wiki states that microformats are

1. a way of thinking about data;
2. design principles for formats;
3. adapted to current behaviors and usage patterns;
4. highly correlated with semantic xhtml;
5. a set of simple data formats that many are actively developing and implementing.





To explain it by example, say you are putting a set of presentation slides online. What format do you use? A proprietary binary format? Some kind of 'SlideShowML' designed by committee? Or a directed subset of XHTML? The latter choice is best described as a microformat, and has been implemented as S5 (A Simple Standards-Based Slide Show System) — a bit more on that later.

Again summarizing the wiki, microformats are, just as importantly, defined by what they are not:

1. not a new language;
2. not infinitely extensible and open-ended;
3. not an attempt to get everyone to change their behavior and rewrite their tools;
4. not a whole new approach that throws away what already works today;
5. not a panacea for all taxonomies, ontologies, and other such abstractions;
6. not defining the whole world, or even just boiling the ocean.

Microformats take aim at a specific problem and solve it, a highly vertical approach. When written up, specifications for microformats are short and readable to the point that it's almost a pleasure to read them.

Does this mean that broadly applicable or horizontal formats are past their prime? Is this an exclusive new way of looking at the world? As much as I'd like to see fewer phonebook-sized specifications, that's not the case. Adam Rifkin's blog mentions "paving the cowpaths", a philosophy that builds on existing infrastructure. Horizontal formats (megaformats?) are a critical substrate underneath microformats. In particular, XHTML appears time and again in a symbiotic relationship with microformats.

Still, some gray areas remain. For example, is RSS a microformat? It seems to bear at least some of the characteristics of one, but on the other hand, a directed subset of XHTML could easily fulfill a similar role in a much more microformat-ish way.

So, we'll have to settle for a slightly vague definition of "microformat". That doesn't mean they're not useful, though, as we shall see.

In Practice

Some of the simplest microformats consist of new values for the existing `rel` attribute. Creative Commons helped inspire `rel='license'`, accompanied by a `href` attribute pointing to an actual license. Also, more weblog authors are starting to use `rel='tag'` to provide folksonomy tagging to their entries. The nofollow was discussed earlier, and still more are in use, including XFN and VoteLinks. All in all, an impressive level of utility for what amounts to reusing an existing attribute.

A more substantial example is XOXO, or Extensible Open XHTML Outlines. Looking at some samples, the markup is stark. At first glance, a reader might feel like something is missing. At a second glance, a reader might appreciate the bare elegance of the format.

An XOXO fragment

```
<ol class='xoxo'>
  <li>Subject 1
    <ol>
        <li>subpoint a</li>
        <li>subpoint b</li>
    </ol>
  </li>
  <li>Subject 2
    <ol compact="compact">
        <li>subpoint c</li>
        <li>subpoint d</li>
    </ol>
  </li>
</ol>
```

Some people might feel warmer and fuzzier with elements named `outline`, `topic`, `item`, and so on, or with elements in a freshly minted namespace, but microformats can still claim the semantic high ground, even when reusing XHTML. In the above, the parts of an outline *are* ordered lists and list items, exactly as the XHTML element names say.

Other key microformats along these lines are hCard (vCard in XHTML) and hCalendar (RFC 2445 iCalendar in XHTML). Eric Meyer's S5 is another great example; if you look behind the scenes, you see nothing but clean, semantically pure markup.

So what are the advantages of reusing XHTML rather than creating from scratch?

1. XHTML has been thoroughly debated, designed, and tested, and mostly avoids pitfalls that a fresh language might step in.
2. Fragments of the microformat can be placed directly in XHTML web pages.
3. Existing tools Just Work.
4. Well-known element names, augmented with judicious `class` and `id` attributes, provide hooks for CSS styling. Default stylesheets do the rest.
5. Additional functionality can be gracefully added within the existing semantics.

Reusing existing markup languages this way is another case of what I call "intentional markup" — that the author's job is to encode her intent through markup, and the client software's job is to render that intent in whatever way makes sense for the local user. Capture the intent using horizontally-standardized markup, and the rest will follow.

Roll Your Own Microformat

Previously, XML-Deviant discussed certification tests, which run on a software platform that looks like it hasn't been updated in five or ten years. The format of the questions is multiple choice, with an indication of whether one or a specific number of answers are correct. This is a well-worn cow path, so let's pave it. I dub the microformat sketched out below the Exam Markup Language, or Examl for short.

The forms language in XHTML 1.x turns out to be slightly clumsy for this sort of thing, so Examl is based on XHTML 2.0, which includes XForms.

An Examl fragment

```
<select ref="q1" class="examl">

  <label>What is the airspeed velocity of an unladen swallow? (pick 2)</lab

  <item>

    <label>I don't know</label>

    <value>1</value>

  </item>

  <item>

    <label>African or European?</label>

    <value>2</value>

  </item>

  <item>

    <label>9.8 m/s*s (for ex-swallows)</label>

    <value>3</value>

  </item>

  <item>
```

```
    <label>Three!</label>

    <value>4</value>

  </item>

</select>
```

Again, the markup clearly expresses intent; CSS, script, or XForms bindings, can be used to refine the user experience to any desired degree.

The above is only a quick sketch. If enough community interest exists, I will develop this language further; watch here for details and announcements. I encourage others to experiment with microformats and post your results.

Are microformats here to stay? The name may be new, but the general idea has been around for almost as long as markup languages have been. The XML ecosystem is expansive, in that lots of new ideas (or recycled old ideas) are continually being explored. If it can be done, someone, somewhere, is probably doing it. In this light, microformats are a refreshing new way of looking at tools we regularly use.

Births, Deaths, and Marriages

Also in XML-Deviant

The More Things Change

Agile XML

Composition

Apple Watch

Life After Ajax?

Tiger XSLT Mapper 2.0

> Free developer edition now available as an Eclipse plug-in.

Saxon dot NET 1.0 RC1

> Now available for download. A few minor changes still expected before final release.

Documents and Data

More on XRules: detailed discussion of using XForms to accomplish similar goals.

Elements vs. attributes, again. Len points to the definitive answer.

Perma-perma-thread.

My new favorite XSLT resource: minesweeper