12 CASE STUDY: YAHOO

In the last chapter, we examined Cork'd, a small, innovative, recent startup. In this chapter, we move on to take a look at a company that represents the other extreme on the Web: Yahoo, arguably the Web's most visited site. Unlike many other highly visited sites, such as Google, Yahoo's traffic comes largely from the content it publishes across a broad spectrum of sites. Yahoo is almost certainly the largest publisher on the Web in this respect.

As you've seen throughout the book, Yahoo uses a range of microformats extensively. In this chapter, we'll examine some of these uses in more detail and talk to a Yahoo developer about the company's interests in microformats, the benefits seen from adopting them, and the lessons learned along the way.

Introducing Nate Koechley

Nate Koechley, Senior Engineer and Technical Evangelist on the Yahoo User Interface (YUI) Library team, generously agreed to talk with me about how Yahoo is using microformats. This is just one of many examples of an openness at Yahoo that I really admire—the company opens up not only APIs and data for any web developer to use, but also a lot of underlying code.

> *Take a look at the YUI Library for articles, documentation, and open source user interface components you can use in your development:* `http://developer.yahoo.com/yui`.

I asked Nate why such an established, huge publisher like Yahoo might be interested in a new technology like microformats.

Why adopt microformats? First and foremost, at Yahoo! we're just plain passionate about the Web. We want to see it thrive. Microformats seem good for the vitality of the Web, and ultimately good for users. It's true there are sometimes constraints on bigger websites and companies, but at the same time our size can be an advantage. In cases like microformats, our scale can help technology reach its tipping point. We're proud to play that role, and look forward to microformats making the Web even more functional for users.

Did Yahoo consider that it was taking risks by doing so?

In terms of risks, people sometimes wonder if microformats void copyright. After speaking with Tantek Çelik, I believe the answer is no, and that the format of data doesn't impact the ownership of data. That said, we're still examining these questions on a case-by-case basis, and we're being cautions in a few cases. (Disclaimer: I am not a lawyer. This statement has not been evaluated by any lawyer. This is NOT legal advice or legal counsel.)

On balance, however, I wish more equations took the "easy to implement + low risk + upside for users + good for the Web's health" format.

How does Yahoo use microformats?

We use microformats all over the place, and more every day. Flickr and Upcoming have been fortified with microformats for longer than I can remember. Yahoo! Tech launched in May with hReview, and in June Yahoo! Local added hCards, hCalendars, and hReviews. You can find microformats on Yahoo! Food, Finance, Movies, Games, and probably many more. I believe our sites in Europe are rolling them out even faster than here in the States.

In addition to the formats I listed, we're also using the hAtom, vCard, and adr formats, and the home link and tag patterns. We have some pretty cool new implementations baking in the oven, but unfortunately I can't talk about them quite yet.

What benefits have you seen from using these microformats? What unexpected benefits?

I think the benefits of microformats haven't been fully realized yet, and won't really until the network effect hits. I believe that day draws near. That said, a developer needs to mark up content in some fashion no matter what, so it's often pretty convenient that we can turn to microformats for guidance.

I wondered, too, whether Yahoo had encountered any difficulties using microformats. The company has adopted several microformats across a broad range of sites, so if anyone was going to run into issues caused by using microformats, it was going to be Yahoo. When I asked about that, Nate had the following to say:

There haven't been too many gotchas. In one case, one of our site's modular template systems made it difficult to associate the product name with the review, but these are solvable problems. Also, some of the formats aren't fully cooked yet, so in some cases there are gaps that I expect will get closed over time. In general, it's been pretty painless.

In the sections that follow, we'll take a look at some of Yahoo's microformat implementations in action, and learn from the experience gained and efforts made by Yahoo's developers. But first, I'd like to emphasize that Yahoo is an early adopter (and in part developer) of microformats. There is a real danger in early adoption of changing specifications. (I know this firsthand—years ago I implemented a draft of CSS2 in Style Master, a CSS editor I develop, and a good deal changed between that draft and the final specification, necessitating a lot of additional work.) In this chapter's case study, you'll see some uses of microformats at Yahoo that aren't 100% "correct," but much if not all of that is a consequence of being an early adopter of changing specifications, or being a true pioneer going into uncharted territory. More than anything, we should be thankful that Yahoo has taken the risks, and appreciative that the company is willing to share its experiences—I know I certainly am.

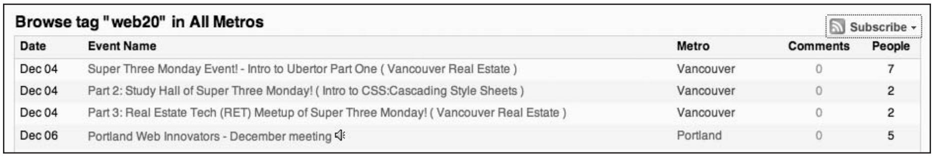
Upcoming

Acquired by Yahoo in late 2005, Upcoming (`http://upcoming.org`) hosts details about tens of thousands of events, and lets users find and track events, and build communities around events (by letting others know they'll attend, or by adding comments about an event). It features all the expected Web 2.0 goodness—an API for creating mashups, event tagging (by organizers and by others using the rel-tag microformat, no less) and, of course, RSS for subscribing to an event or a collection of events (e.g., all events tagged "web2.0", all events in Vancouver or some other area, etc.). It should come as no surprise that Upcoming events are marked up using the hCalendar microformat. In fact, Upcoming doesn't use hCalendar just on an event's page, as you would expect, but also when presenting lists of events, like all those with a given tag or in a given place.

In the sections that follow, we'll look at an example of Upcoming's use of hCalendar and explore definition lists in more detail.

hCalendar on Upcoming

In Figure 12-1, I've subscribed to all the events tagged "web2.0" (everything about the Web is now tagged "2.0" it seems—even events for realtors).



Browse tag "web20" in All Metros

Date	Event Name	Metro	Comments	People
Dec 04	Super Three Monday Event! - Intro to Ubertor Part One (Vancouver Real Estate)	Vancouver	0	7
Dec 04	Part 2: Study Hall of Super Three Monday! (Intro to CSS:Cascading Style Sheets)	Vancouver	0	2
Dec 04	Part 3: Real Estate Tech (RET) Meetup of Super Three Monday! (Vancouver Real Estate)	Vancouver	0	2
Dec 06	Portland Web Innovators - December meeting 🔈	Portland	0	5

Figure 12-1. Search results for Web 2.0 events in Upcoming

Reproduced with permission of Yahoo! Inc. © 2007 by Yahoo! Inc. YAHOO! and the YAHOO! logo are trademarks of Yahoo! Inc.

Let's take a look at the code underneath:

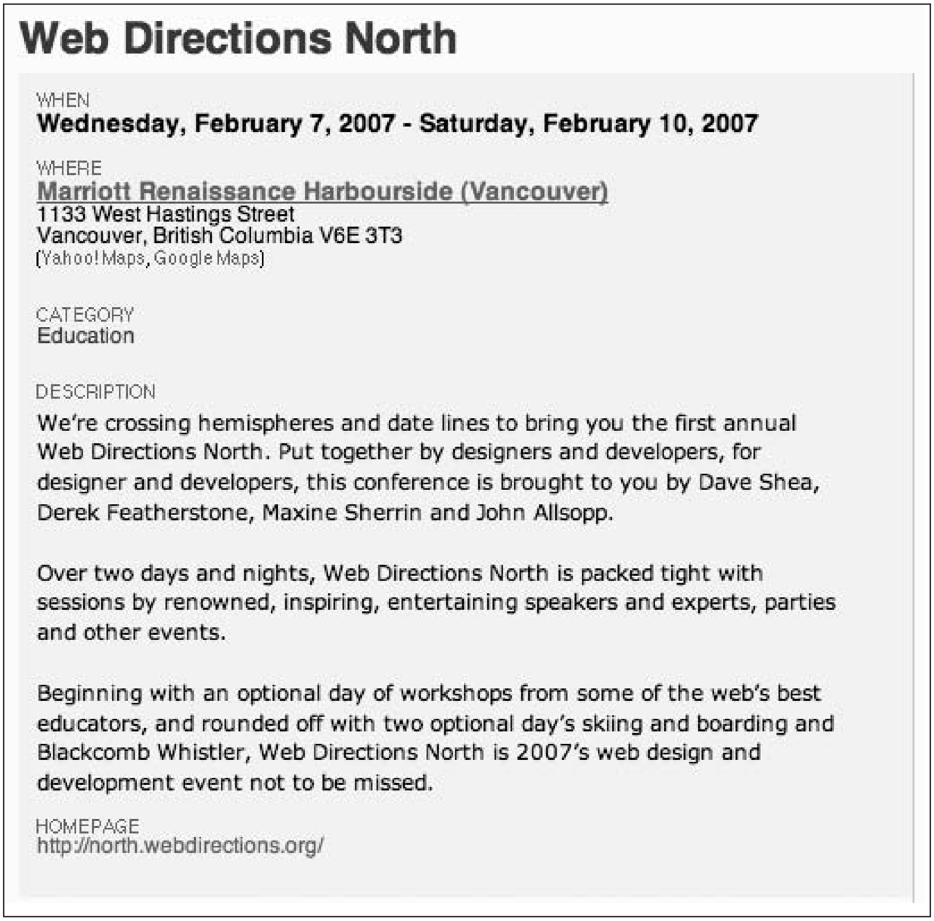
```
<tr class="vevent">
  <td class="even" nowrap="nowrap"><abbr class="dtstart" title=""> ➥
    Dec 04</abbr></td>
  <td class="even"><a href="http://upcoming.org/event/118987" ➥
    class="url summary" >Super Three Monday Event! - Intro to Ubertor ➥
    Part One ( Vancouver Real Estate )</a></td>
  <td class="even"><a href="/metro/ca/bc/van/"><abbr class="location" ➥
    title="Ubertor @ Milan&#039;s Bistro, 1223 Hamilton Street, ➥
    Vancouver, V6B6A8">Vancouver</abbr></a></td>
  <td class="even" align="center"><span class="inactive">0</span></td>
  <td class="even" align="center">7</td>
</tr>
```

As you can see, it's not quite perfect yet—for example, the start date has a human-friendly component, but not a machine-readable part, which makes it far less valuable than if that machine-readable date was there. Upcoming has also interestingly used an abbr design pattern to give the location details:

```
<abbr class="location" title="Ubertor @ Milan&#039;s Bistro, 1223 ➥
  Hamilton Street, Vancouver, V6B6A8">Vancouver</abbr>
```

Keep in mind that the page is a summary, and as such space is at a premium. As a result, an additional usability benefit in most browsers is that by hovering over the element, the reader gets the full details of the location. It's a feature users will need to learn, but if they use the site frequently, it might start coming in very handy, because if they just want a brief overview of the event they no longer need to follow the link to the full details page, saving a click.

If users do want more details, they can follow the link and get a more detailed hCalendar, as shown in Figure 12-2.



Web Directions North

WHEN
Wednesday, February 7, 2007 - Saturday, February 10, 2007

WHERE
Marriott Renaissance Harbourside (Vancouver)
1133 West Hastings Street
Vancouver, British Columbia V6E 3T3
(Yahoo! Maps, Google Maps)

CATEGORY
Education

DESCRIPTION
We're crossing hemispheres and date lines to bring you the first annual Web Directions North. Put together by designers and developers, for designer and developers, this conference is brought to you by Dave Shea, Derek Featherstone, Maxine Sherrin and John Allsopp.

Over two days and nights, Web Directions North is packed tight with sessions by renowned, inspiring, entertaining speakers and experts, parties and other events.

Beginning with an optional day of workshops from some of the web's best educators, and rounded off with two optional day's skiing and boarding and Blackcomb Whistler, Web Directions North is 2007's web design and development event not to be missed.

HOMEPAGE
http://north.webdirections.org/

Figure 12-2. Further details on each event are obtained by following the relevant links.

The hCalendar has this HTML (I’ve elided it a little to make it more readable):

```
<div id="event" class="vevent">
  <h1 class="name summary">Web Directions North</h1>
  <div id="eventMain">
    <div id="eventMetadata">
      <div class="small">When</div>
      <div class="date">
        <abbr class="dtstart" title="2007-02-07">Wednesday, ➥
          February 7, 2007</abbr> - <abbr class="dtend" title="2007-02
          10">Saturday, February 10, 2007</abbr>
      </div> <!-- /.date -->
      [...]
      <div class="venue location vcard">
        <br/><div class="small">Where</div>
        <span class="fn org"><a href="/venue/35093">Marriott ➥
          Renaissance Harbourside (Vancouver)</a></span><br />
        <div class="address adr">
          <span class="street-address">1133 West Hastings ➥
            Street</span><br />
          <span class="locality">Vancouver</span>, <span ➥
            class="region">British Columbia</span> <span ➥
            class="postal-code">V6E 3T3</span>
        </div>
        <div class="directions">
          (<a href="http://maps.yahoo.com/...6E+3T3">Yahoo! Maps</a>,
          <a href="http://maps.google.com/maps?q=1133+West+...3"> ➥
            Google Maps</a>)<br />
        </div>
      </div> <!-- /.venue -->

      <div class="category">
        <div class="small">Category</div>
        Education
      </div>

      <div class="description">
        <div class="small">Description</div>
        We're crossing hemispheres ...
      </div>

      <div class="url">
        <div class="small">Homepage</div> <a ➥
          href="http://north.webdirections.org/"> ➥
          http://north.webdirections.org/</a>
      </div>

      [...]

</div> <!-- /#eventMain -->
```

One "gotcha" you've already seen with dates in hCalendar is evidenced here: the dtend property has a value of title="2007-02-10", which as you might recall actually means in effect midnight between February 9 and 10. To get around this nonintuitive aspect of hCalendar, a finish time with the date such as 2007-02-10T1700-0800 or the date value of 2007-02-11 would correctly specify the machine-readable date.

We can see how the microformat markup has most likely been added to the existing markup, for example:

```
<div class="venue location vcard">
```

and

```
<div class="address adr">
```

Commenting is something that many web developers don't pay enough attention to. Particularly with many nested elements like the divs here, knowing where one ends can be difficult to work out, so adding comments like this makes the markup much more readable and maintainable:

```
</div> <!-- /#eventMain -->
```

There are some little things that might need some fixing up, for example:

```
<div class="url">
  <div class="small">Homepage</div> <a ➥
    href="http://north.webdirections.org/"> ➥
    http://north.webdirections.org/</a>
</div>
```

Because the div with the class of url contains the element <div class="small">Homepage </div> as well as the link, the actual value of the URL includes the text Homepage. Perhaps better markup might be as follows:

```
<dl>
  <dt>Homepage </dt>
  <dd><a href="http://north.webdirections.org/" class="url"> ➥
    http://north.webdirections.org/</a></dd>
</dl>
```

In fact, while not strictly associated with microformats, this pattern of a definition list could replace a number of multiple div constructs. For example, the following:

```
<div class="category">
  <div class="small">Category</div>
  Education
</div>
```

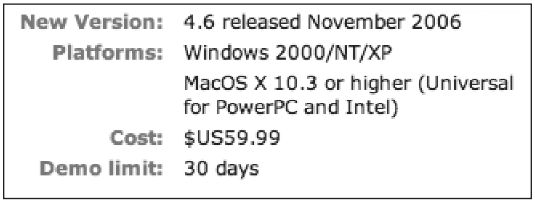
could be more cleanly marked up as follows:

```
<dl>
  <dt class="small">Category</dt>
  <dd class="category">Education</dd>
</dl>
```

In the original code, the value of the `category` property is in fact `Category Education`. But what we really want is for the value of the `category` property to simply be `Education`, which it is in the reformulation.

Definition lists

If you haven't used **definition lists** before (most developers use them rarely if at all), they are actually more broadly useful than is assumed or their name would imply. According to the HTML specification, "Definition lists vary only slightly from other types of lists in that list items consist of two parts: a term and a description." So definition lists can be used in not uncommon situations like the example shown in Figure 12-3 (from my website), where I have labeled text entries.



New Version:	4.6 released November 2006
Platforms:	Windows 2000/NT/XP
	MacOS X 10.3 or higher (Universal for PowerPC and Intel)
Cost:	$US59.99
Demo limit:	30 days

Figure 12-3. A good use for the noble definition list

This list is marked up as follows in HTML:

```
<dl class="prod-summary">
  <dt>New Version:</dt>
  <dd>4.6 released November 2006</dd>
  <dt>Platforms:</dt>
  <dd>Windows 2000/NT/XP</dd>
  <dd>MacOS X 10.3 or higher (Universal for PowerPC and Intel)</dd>
  <dt>Cost:</dt>
  <dd>$US59.99</dd>
  <dt>Demo limit:</dt>
  <dd>30 days</dd>
</dl>
```

In the preceding code, the `<dt>` elements are floated to the left to put the `<dt>` and its associated `<dd>` element on the same line.

As for CSS, Upcoming doesn't really use the hCalendar markup for styling with CSS. That would make sense where the HTML has had the microformat markup retrofitted to it.

Nate more or less confirmed my speculation in an email when he commented, “Your hunches about legacy impact were basically spot on.”

Reviews at Yahoo Local

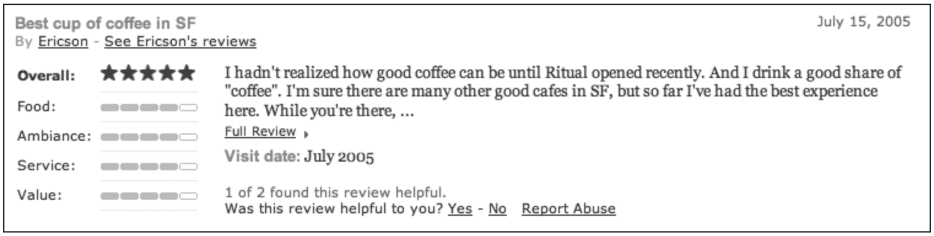
Review sites seem to be the flavor of the month, and as you learned earlier, Yahoo uses hReview at a couple of its very popular sites.

Yahoo Local embodies the “user-generated content” nature of many recent sites. The idea is to let users add their opinions and expertise to a site, in order to tap into the “wisdom of the crowd.” At present, much of this content is still centralized at sites like Cork’d and Yahoo Local, but as you’ve seen on several occasions, one of the goals of microformats is to enable decentralized services. By adopting hReview, Yahoo makes it possible for others to aggregate the reviews they publish much more easily than would otherwise be possible.

In the following sections, we’ll explore Yahoo Local’s use of hReview and some lessons we can learn from that.

Review example

So what does a review look like at Yahoo Local? Figure 12-4 shows the review page for a well-liked coffee shop in San Francisco.



Best cup of coffee in SF
By Ericson - See Ericson's reviews

July 15, 2005

Overall:	★★★★★
Food:	▬▬▬▬▭
Ambiance:	▬▬▬▬▭
Service:	▬▬▬▬▭
Value:	▬▬▬▬▭

I hadn't realized how good coffee can be until Ritual opened recently. And I drink a good share of "coffee". I'm sure there are many other good cafes in SF, but so far I've had the best experience here. While you're there, ...

Full Review ▸

Visit date: July 2005

1 of 2 found this review helpful.
Was this review helpful to you? Yes - No Report Abuse

Figure 12-4. A review from Yahoo Local

Reproduced with permission of Yahoo! Inc. © 2007 by Yahoo! Inc. YAHOO! and the YAHOO! logo are trademarks of Yahoo! Inc.

Stop for a moment and think of how you would mark this up. What hReview properties are there? Now, how does Yahoo Local do it? Let’s have a look (I’ve elided the code that is not directly relevant to us):

```
<tbody class="hreview">

  <tr valign="top"><th colspan="2"><h3><u style="float:right;"><abbr ➥
    style="border-bottom-style: none;" class="dtreviewed" ➥
    title="2005-07-15">July 15, 2005</abbr></u><span class="summary"> ➥
    Best cup of coffee in SF</span></h3><h6 class="reviewer vcard">By ➥
    <a ...>Ericson</a> - <a href="..." class="url">See Ericson's ➥
    reviews </a></h6></th></tr>
  <tr valign="top"><td width="170" nowrap><table class="ylsrevrating">

  <tr valign="middle"><th><b>Overall: </b></th><td><abbr ➥
    style="border-bottom-style: none;" class="rating" title="5"><img ➥
    src="http://us.i1.yimg.com/us.yimg.com/i/us/ls/gr/ ➥
    read_star_5.gif"alt="5" width="78" height="13"></abbr></td></tr>

  <tr valign="middle"><th>Food:</th><td><abbr ➥
    style="border-bottom-style: none;" class="rating" title="5"><img ➥
    src="http://us.i1.yimg.com/us.yimg.com/i/us/sh/karma/ ➥
    ur_bar_4.gif"alt="4" width="79" height="6"></abbr></td></tr>

  <tr valign="middle"><th>Ambiance:</th><td><abbr ➥
    style="border-bottom-style: none;" class="rating" title="5"><img ➥
    src="http://us.i1.yimg.com/us.yimg.com/i/us/sh/karma/ ➥
    ur_bar_4.gif"alt="4" width="79" height="6"></abbr></td></tr>

  <tr valign="middle"><th>Service:</th><td><abbr ➥
    style="border-bottom-style: none;" class="rating" title="5"><img ➥
    src="http://us.i1.yimg.com/us.yimg.com/i/us/sh/karma/ ➥
    ur_bar_4.gif"alt="4" width="79" height="6"></abbr></td></tr>

  <tr valign="middle"><th>Value:</th><td><abbr ➥
    style="border-bottom-style: none;" class="rating" title="5"><img ➥
    src="http://us.i1.yimg.com/us.yimg.com/i/us/sh/karma/ ➥
    ur_bar_4.gif" alt="4" width="79" height="6"></abbr></td></tr> ➥
    </table></td><td? ➥
    width="100%"><p class="ylsratrevfull" id="full_4"><span ➥
    class="description">I hadn&#39;t realized how good coffee can be ➥
    until Ritual opened recently. ...<a ...>Full Review</a><script> ➥
    revMin("4");</script></p>...</td></tr>

</tbody>
```

The first thing to note is that the review is a table. While the knee-jerk response might be to say, "Hey, `that's` not tabular data!", I think it is fair to argue that it is. Whether I'd recommend it be marked up that way is a separate matter—some data is clearly of a tabular nature, and tables are ideal for marking them up, but other data is less clearly so. But if we delve more deeply, we'll find that there are in fact two tables at play.

Inside each review is a table of the ratings. To me, this is an excellent use of tabular markup. We have five rows of related data—`Overall`, `Food`, `Ambience`, `Service`, and `Value`—and for each of these, a rating.

What about the other table? Well, from the example I've pulled out here, you might think that the review itself is in a table. But in fact it's in a table body element (`<tbody>`).

If you've used `<tbody>` elements, you might assume that every table has at most one `<tbody>`, but a table may have any number of `<tbody>` elements. If we zoom out just a fraction, we find the following structure for the collection of reviews:

```
<table ... class="ylsreviewrow">
  <tbody class="hreview">
    ...
  </tbody>

  <tbody class="hreview">
    ...
  </tbody>

  <tbody class="hreview">
    ...
  </tbody>

</table>
```

Each review is not in a table to itself; rather, it's in a `<tbody>` for the set of reviews, which is in a table. We can think of the set of reviews as a one-dimensional table of reviews. There are certainly other ways this set of reviews could be marked up, but of course that's one of the beauties of HTML, and indeed microformats.

It could be a list—after all, a one-dimensional table is in many ways a list. Or, each review could be contained within a div. I suspect—though it's purely speculation on my part—that the choice of markup is a complex one involving legacy solutions. For example, the reviews page allows sorting of the reviews by newest, oldest, most helpful, and so on, and there may be constraints imposed by the server-side application providing this functionality.

None of my commentary here should be construed as criticism, by the way. Rather, I want to demonstrate that while sometimes we have the luxury of the "ideal" or (arguably) "perfect" solution, in reality, we are very often constrained by factors such as the following, which make solving a problem much harder:

- Time
- Resources
- Legacy systems

I think Yahoo Local has done an admirable job of incorporating microformats into a complex existing service and using HTML appropriately despite considerable constraints.

Sometimes as developers we are fortunate that we can build a brand-new system from the ground up. But frequently we'll be maintaining and upgrading existing systems. That's when our job often becomes much more difficult.

Something specific worth noting about Yahoo Local's use of hReview is that multiple ratings are used—not just an overall rating, but individual ratings for specific aspects of the experience, like ambience and service. And Yahoo has marked these up using the abbr design pattern, just as Cork'd marks up ratings. In both cases, rather than a textual, numerical value, the reviews use a graphical indicator—stars in the case of Cork'd and bars in the case of Yahoo Local. The use of a table in this case actually improves accessibility, as screen readers can read out the <th> and <td> elements.

Styling Yahoo Local reviews

One indicator that the hReview microformatting on Yahoo Local has been retrofitted is that little if any of the page's styling is done using the hReview microformat. Again, this demonstrates the flexibility of microformats—they can often be added to even very complex, longstanding applications, without any significant impact on the existing code. Here's something you might have noticed in the code from the previous section:

```
<abbr style="border-bottom-style: none;" class="rating" title="5">
```

Why this inline style? Some browsers add dotted underlining via the CSS border property to the <abbr> element as part of their default style sheets. This inline CSS overrides that styling. Ideally, rather than add the inline style to each <abbr> element, we'd simply have a CSS rule:

```
abbr {border-bottom-style: none}
```

In fact, when I raised this with Nate Koechley, he mentioned that this statement is part of the YUI Reset CSS file (see http://developer.yahoo.com/yui/reset), which "normalizes the default rendering of all HTML elements, for example it sets margin, padding, and border to 0, font sizes to [a] default, italic and bold styles to normal, and list-style to none."

However, at present, Yahoo Local doesn't use Reset CSS, and I imagine there are legacy or perhaps internal procedural implications that make simply adding the inline style a much easier and possibly shorter-term solution.

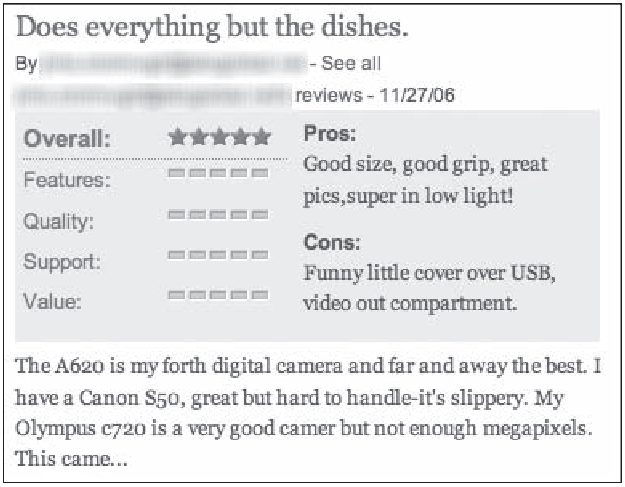
hReviews at Yahoo Tech

Let's compare Yahoo Local's reviews with those over at Yahoo Tech. Interestingly, Yahoo Tech uses hReview for the summary page of reviews of a particular product, but not for the full review page by an individual reviewer.

In the following sections, we'll see how Yahoo Tech uses hReview, and how this differs from Yahoo Local.

Review example

Figure 12-5 shows what a Yahoo Tech review looks like in a browser.



Does everything but the dishes.

By [redacted] - See all [redacted] reviews - 11/27/06

Overall:	★★★★★	Pros:
Features:	▭▭▭▭▭	Good size, good grip, great pics,super in low light!
Quality:	▭▭▭▭▭	
Support:	▭▭▭▭▭	Cons:
Value:	▭▭▭▭▭	Funny little cover over USB, video out compartment.

The A620 is my forth digital camera and far and away the best. I have a Canon S50, great but hard to handle-it's slippery. My Olympus c720 is a very good camer but not enough megapixels. This came...

Figure 12-5. A review from Yahoo Tech

Reproduced with permission of Yahoo! Inc. © 2007 by Yahoo! Inc. YAHOO! and the YAHOO! logo are trademarks of Yahoo! Inc.

In many ways it's the same as (or similar to) a review at Yahoo Local. But let's take a look at the HTML (again, I've elided some code to get to the most relevant parts for our purposes):

```
<div class="arating hreview">
  <div class="hd">
    <h3 class="summary"><a ...</a></h3>
    <p class="metadetails ">By <a href="..."  ... class="reviewer ➥
      fn"> reviewer@obfuscated.com</a> ... - <span class="dtreviewed"> ➥
      11/27/06</span><a href="#prodname" class="include ➥
      microformatdetail"></a></p>
  </div>

  <div class="bd">
    <div class="ratreviewsummary">
      <ul class="ratingslist">
        <li class="overall stars10 ">Overall: <span ➥
          title="reviewer@obfuscated.com gave this product 5 out ➥
          of 5 stars for Overall quality" class="rating"> ➥
          <em class ="value">5</em>/<em class="best">5</em></span>
        </li>
        <li class="bars10">Features: <span ➥
          title="reviewer@obfuscated.com gave this product 5 out ➥
          of 5 stars for Features" class="rating">5/5 </span>
        </li>
        <li class="bars10">Quality: <span ➥
          title="reviewer@obfuscated.com gave this product 5 out ➥
          of 5 stars for Quality" class="rating">5/5 </span>
        </li>
        <li class="bars10">Support: <span ➥
          title="reviewer@obfuscated.com gave this product 5 out ➥
          of 5 stars for Support" class="rating">5/5 </span>
        </li>
        <li class="bars10">Value: <span ➥
          title="reviewer@obfuscated.com gave this product 5 out ➥
          of 5 stars for Value" class="rating">5/5 </span>
        </li>
      </ul>

      <dl class="procons description">
        <dt>Pros:</dt>
        <dd>Good size, good grip, great pics, super in low light!</dd>
        </dl>
        <dl class="procons description ">
        <dt>Cons:</dt>
        <dd>Funny little cover over USB, video out compartment.</dd>
      </dl>
    </div>
    <div class="ytuserreviewtext">
      <div class="ytReviewContainer">
        <p class="description item fn">The A620 is my forth digital ➥
          camera and far and away the best. I have a Canon S50, ➥
          great but hard to handle-it's </p>
      </div>
    </div>
  </div>
</div>
```

This is much closer to the div- and class-based markup we might usually expect. Unordered lists for ratings and definition lists for marking up pros and cons are used as well.

But there are some little issues that need fixing. Again, I don't mean to be critical—many of these things are still evolving, and indeed it's most likely that this code was implemented before hReview got to its current state.

First, while the overall rating is correctly marked up, as follows:

```
<li class="overall stars10 ">Overall: <span ➥
  title="reviewer@obfuscated.com gave this product 5 out of 5 ➥
  stars for Overall quality" class="rating"><em class="value">5 ➥
  </em>/<em class="best">5</em></span>
```

The individual ratings are not quite right. Instead of this:

```
<li class="bars10">Features: <span ➥
  title="reviewer@obfuscated.com gave this product 5 out of 5 ➥
  stars for Features" class="rating">5/5 </span>
```

where the value is 5/5, but should be a numerical value to a single decimal place, we could use the same pattern as for the overall ratings like so:

```
<li class="bars10">Features: <span ➥
  title="reviewer@obfuscated.com gave this product 5 out of 5 ➥
  stars for Features" class="rating"><em class="value">5 ➥
  </em>/<em class="best">5</em></span>
```

Or, because the default “best” is 5, we could simply use the following:

```
<li class="bars10">Features: <span ➥
  title="reviewer@obfuscated.com gave this product 5 out of 5 ➥
  stars for Features" class="rating">5</span>/5
```

Another important thing to note is that the FN for an item *must* appear inside the element for the item, not as a class value on the same element that has item as a class value.

Here, we'll need to make a small change, because for now, the fn for the item is on the item element, not on a descendent element:

```
<p class="description item fn">The A620 is my forth digital camera ➥
  and far and away the best. I have a Canon S50, great but hard to ➥
  handle-it's ...</p>
```

Also, while it is not an invalid hReview to have both a description and item for the same element (i.e., “The A620 is my fourth digital camera . . .”), this is not really the item we are reviewing. So, here it would be best to remove the item and fn class values from the description paragraph:

```
<p class="description">The A620 is my forth digital camera and ➥
  far and away the best. I have a Canon S50, great but hard to ➥
  handle-it's ...</p>
```

But we still need an item for the review. Remember, an item property is *required* in hReview. There is more than one review on this page of the same item, so that might bring to mind the include design pattern discussed in Chapter 9. We can use this design pattern to include details from one microformat to another, essentially when the microformats represent the same thing.

12

The first step is to uniquely identify the microformatted element we want to include a property or properties from. In this case, we’ll have to look outside the hReview element itself for that element. If we look at the page itself, we find the following HTML:

```
<strong class="item fn" id="prodname">Canon PowerShot A620</strong>
```

We still have the problem that the fn for the item appears as a class on the element for the item, rather than on a descendent element for the item element. So, we change it as follows:

```
<strong class="item" id="prodname"><span class="fn">Canon PowerShot ➥
  A620</span></strong>
```

We give the item a unique identifier, using the id. Now we can return to our individual review, and include this in there:

```
<p class="description">The <span class="item"><a href="#prodname" ➥
  class="include"></a>A620</span> is my forth digital camera and ➥
  far and away the best. I have a Canon S50, great but hard to ➥
  handle-it's ...</p>
```

If we look back to the top of the review, we find something quite similar to this:

```
<a href="#prodname" class="include microformatdetail"></a>
```

A couple of other little things need attention. For example, a reviewer, if included, *must* be an hCard:

```
By <a href="..."  ... class="reviewer fn">reviewer@obfuscated.com</a>
```

Here, we need to add a class value of vcard and then make the fn on a descendent element of that element (remember the properties of a compound microformat are descendents of the root element). We end up with something like this:

```
By <span class="reviewer vcard"><a href=".."class="fn"> ➥
  reviewer@obfuscated.com</a></span>
```

Styling Yahoo Tech reviews

As with Yahoo Local, the Yahoo Tech site doesn't use the microformat markup for styling with CSS. Once more, I suspect it's because the markup has had microformats retrofitted onto it, so styling was already well and truly in place with the existing markup.

Is it worth it?

Having seen how Yahoo is adopting microformats in some pretty significant ways, a question comes to mind: is it worth it? I next asked Nate whether he felt adopting microformats has been worth the effort.

Anything that makes the Web a better place feels like the right thing to do. When it's as technically low-impact as microformats, why not? Each decision needs to be considered for each new context, of course, but implementing microformats seems like an easy win. That our efforts might positively impact the overall evolution of the Web is a bonus.

Finally, I asked Nate if he could tell us where Yahoo might go next with microformats.

I can't talk about future plans, but I will say that many people here are pretty excited about microformats, and some are actively involved with microformats.org. I think we've already demonstrated our willingness to deploy them, and I think it's safe to expect more of the same from Yahoo!.

Summary

This case study on Yahoo and the previous one on Cork'd demonstrate the versatility of microformats. New projects like Cork'd can take advantage of microformats from the ground up, not just for markup, but for styling with CSS as well. Existing projects—even large, sophisticated applications like those presented in this chapter from Yahoo—can be adapted to use microformats with little impact on the existing code base, HTML, and CSS.

I hope that seeing how such a broad range of industry leaders have adopted microformats and benefited from doing so will reassure you that microformats aren't just great bleeding-edge concepts, but a practical, useful, and increasingly commonly used technology. I hope, too, that these case studies inspire you to start using microformats in your existing and new projects.

In the next and final chapter, you'll see how new microformats come into being. Not only is this of interest if you want to be involved in the development of new microformats, but also the process that has developed is applicable if you want to standardize internal markup conventions for your company or develop specific microformat-like solutions to common development problems you face.