Emerging Web technologies

A look ahead

Martin Streicher

If the Web were a celebrity, the top tabloid headlines of 2009 might include, "Cloud Computing Parties All Night, Snogs Servers"; "Aliens Twitter: 'We Come in Peace'"; and "Gosselin in New Reality TV Show: Eight Browsers Is Enough."

Hyperbole aside, cloud computing, real-time content, and advancements in browser speed and capabilities made the Web more compelling than ever before. Robust infrastructure—from Amazon Simple Storage Service (S3) for storage to SendGrid for scalable e-mail—can now be leased with just a few clicks of the mouse. Instant information, characterized by tweets and Flickr updates sent from anywhere at any time, are de rigueur. And the heated competition between Google Chrome, Apple Safari, Mozilla Firefox, and Windows® Internet Explorer® hews the browser closer and closer to a stand-alone platform—and in the case of Chrome, the heart of an entirely new kind of operating system.

Surely, the paparazzi will continue to stalk these stories in 2010, looking to scoop stories and photos of Microsoft's own cloud offering, the travails of Twitter, and the cat fight between browser makers. But other software, developments, and trends are sure to make headlines and seduce developers in 2010. Here's a look at the top stories you can expect to read in technology tabloids.

**HTML's extreme makeover**

It's not just starlets going under the knife. The venerable Hypertext Markup Language is getting a face lift, too. Hardly shocking, the now-decades-old HTML format is struggling to keep pace
with innovation. After all, HTML was invented to structure and package documents, not Web applications.

HTML 5 aims to address the needs of the modern Web, with extensive support for programmatic interaction between content and the local computer. Among other novelties, HTML includes programmatic interfaces (APIs) to draw arbitrary graphics in the new canvas, find your position on the globe, cache code and data, and offload compute-intensive tasks to keep the interactive portion of the browser responsive.

With the advent of video and audio tags, HTML 5 obsolesces the most common uses of Adobe® Flash® and Microsoft® Silverlight™, too, freeing developers and consumers from those proprietary solutions. HTML 5 embeds a movie or soundtrack as simply as HTML 4 embeds an image:

```html
<video src="/video/redcarpet.ogg" width="400" height="300" controls>
```

HTML 5 also formalizes the otherwise ad hoc techniques used to structure content. Common Web page constructs such as `<div class="header">`, `<div class="footer">`, `<div class="article">`, and `<ul class="nav">` are replaced with the obvious `<header>`, `<footer>`, `<article>`, and `<nav>` tags, respectively, among others. The latter, standardized tags imply context, making it easier for spiders and scrapers to tease pertinent content from a page.

Although HTML 5 is not a ratified standard, and there is seemingly no timetable to stamp it "official" (it will be "years" in the making, according to its stewards), most of the leading Web browsers already implement many of its more powerful features. For example, Chrome, Firefox, Safari, and Opera all support the APIs enumerated above, and you can find impressive demonstrations of each online. Moreover, developers are actively creating and deploying applications based on these new capabilities. Prominent examples include Wave, Google's conversation engine now in beta; SproutCore, an application framework based entirely on JavaScript™, HTML 5, and CSS; and Mozilla Lab's Bespin, a rich, collaborative code editor that runs entirely in the browser and saves all your work in a cloud.

HTML 5 is a moving target. That said, expect more applications to take advantage of those portions of the draft specification concretely defined, widely implemented, and stable. Even now, HTML 5 is surprisingly well supported in mobile browsers, including the Apple iPhone and Android. Indeed, mobile is the most likely target for innovative HTML 5 applications.

**Torrid database relationships: "It's not exclusive anymore"**

Web developers have been enamored of the relational database (RDBMS) for a very long time. Perl's DBI, created a decade ago or more, was among the first to add dynamism and personalization to Web pages. But the RDBMS isn't the exclusive choice any more. The once-inseparable LAMP stack is breaking up.

Strictly speaking, an RDBMS akin to Oracle, MySQL, PostgreSQL, and SQLite isn't a Web technology. In fact, it's quite the opposite: a relatively unsexy piece of machinery relegated to the lowest tiers of Web infrastructure. However, the modern Web could hardly function without the countless schemas that persist everything from Amazon's merchandise to Wikipedia's wisdom.
But like HTML, the familiar RDBMS struggles to meet the practical demands of modern Web applications—specifically, not all information is suited to normalization. Documents are one datum typically shoe-horned into rows and columns; irregular data is a mismatch for RDBMSs, as well. Further, terabytes and petabytes of data aren't easily managed in a single system. Unique applications require specialized storage to simplify analysis, too.

Pioneering developers started migrating to so-called "No SQL" solutions within the past 18 months, and the "fad" is now mainstream. With odd names such as MongoDB, BigTable, Cassandra, and Tokyo Cabinet, these stores are designed to shard, span thousands of commodity servers, and embody petabytes of information. Beyond capacity, No SQL software often maps more readily to a problem domain. A document-oriented database such as MongoDB is one example, as is the object-oriented database, which does away with object-relational mapping (ORM) completely. Even the RDBMS is adapting to the realities of the Web's third decade. For example, Drizzle is a fork of MySQL version 6 that's smaller, faster, and simpler, tuned for the Web.

Not too long ago, the obvious answer to persistent storage was the RDBMS. Now, the only obvious answer is, "It depends." Developers are playing the field, looking for a better match.

**Data: caught out in public**

Traditionally, data has skulked about, protected and doled out carefully by its handlers, or has hidden from view, wearing dark sunglasses, mingling among the commoners of the Web. At times, you have to be something of an online, digital "dumpster diver" to find the information you want; if you are lucky, one of the search engines or Wolfram may turn up the tidbit for you. However, even Google and Yahoo! only spider the Web every so often, translating to some delay before pertinent information turns up. In fact, the Web may not always offer the best information. Perhaps the ideal source is a demonstrator at a protest, a blogger at a conference, or a diner sitting in a hot new restaurant.

Mashups hint at the promise of synthesized data—information culled and collated from many sources into a greater whole—and many have achieved critical and popular success. Google Maps is an impressive example, for one, annotating business directories with search results and street and satellite maps. Services like Facebook add social fabric, and the omnipresent Twitter adds immediacy. Yet, each is just a harbinger of what's to come this year, as data of all sorts is assembled instantly by compelling applications. Consider Trapster, a do-it-yourself (literally), real-time speed trap detector; or Layar, a proprietor of layers, or augmentations, that can enhance any outing with walking tours full of facts, figures, and virtual realities.

Arguably, augmentation is more trend than technology, but an effective experience is sure to be composed of many moving parts, such as image and pattern recognition, search, data collection, analysis, dissemination, and more. Geolocation is essential, but so too is social networking. Some information can be based on reference material and traditional sources, but the more personal insights will be those provided by others.
A bankable star

Stop me if you've heard this treatment before: One day, in the not-too-distant future, you'll be able to pay for anything with your digital wallet.

Sure, going cashless is a familiar prognostication, dating back to the dawn of the Web, if not earlier; you may even consider it cliché. Nonetheless, micropayments and electronic transactions are coming in 2010. And it'll turn out better than the Adventures of Pluto Nash.

Until recently, there was too little infrastructure to support micropayments and digital wallets. In the case of the former, few providers, if any, divvied up content a la carte. In the case of the latter, cashless payment methods, including the automated teller machine (ATM) card and the credit card, typically require a cash register and some physical connection to a clearinghouse network. But the combination of the Web and mobile devices—be it an eBook reader, phone, netbook, or laptop—is sufficient infrastructure to effectuate all kinds of transactions, small and large. To wit, consider the iPhone, the Amazon Kindle (also available for the iPhone), and the new Barnes and Noble Nook. Each can make a purchase with the touch of a button. Certainly, the Kindle and Nook are tied to proprietary services, but the iPhone is an exception. Many applications charge micropayments in situ to add levels and add-ons, and Square, which launched at the very end of 2009, permits almost anyone to accept payment via the iPhone. Many retailers and banks provide iPhone applications to streamline transactions.

Amazon's PayPhrase and mobile payments platforms PayPal's PayPalX and Square are leading the charge, so to speak. Developers and entrepreneurs are sure to create business models, processes, and applications to sell products in radically new ways.

It girl caught texting on . . . OMG!

There's something of a revolution going on. After two decades of plodding developments in personal computing, a host of new platforms has cropped up, many within the past six months of 2009. Google confirmed the existence of Nexus One, its own telephone, while Google's Android operating system powers a number of other mobile devices, including the Barnes and Noble Nook. Apple released Mac OS X version 10.6 Snow Leopard in September 2009, while Microsoft proffered Windows 7 shortly after. The iPhone received a much-needed update to hardware and software, while its companion App Store has transformed how software, music, and movies are sold and consumed. Suddenly, it feels like the early days of personal computing, albeit computers are now deeply ingrained in nearly every aspect of life.

The browser is now just one portal into the World Wide Web. Entities no longer have one Web site but a smattering of personalities—one for each kind of device. A traditional UI will suffice for the desktop, but a mobile device demands a more minimalist approach and one fine-tuned to its capabilities, screen size, and typical user. So many variants sounds difficult and expensive to create. However, unlike developing software for the desktop, developers can leverage one set of technology—HTML 5, JavaScript, and CSS—for all platforms.
**The great sequels: Toy Story 3, Rails 3, PHP 6**

In addition to the return of Buzz Lightyear to the big screen, 2010 heralds a number of technology sequels planned for the small screen.

Ruby on Rails version 3, slated for some time mid-2010, merges the popular Rails and Merb frameworks and adds a slew of features designed to speed up both the performance of Rails applications and programmers. The Rails router is faster and can route by sub-domains and to individual pieces of middleware. Rails 3 precludes cross-site scripting (XSS) attacks and abstracts common Ajax techniques, making the resulting JavaScript code unobtrusive and agnostic to JavaScript frameworks.

PHP version 6 integrates Unicode everywhere, adds large numbers, and may include traits, a way to mix in methods to a class, avoiding the all-or-nothing approach of multiple-inheritance. (If you are a Ruby developer, traits echo the capabilities of module and include). Here is an example (provided with permission by Mark Story):

```php
trait Sluggable {
    function slug() {
        return preg_replace('/[\D\d]+/i', '_', strtolower($this->title));
    }
}

class Post extends Model {
    use Sluggable;
}
```

Other influential sequels include Ubuntu version 10.4 (Lucid Lynx); Google Chrome OS; GNOME version 3.0; Drizzle version 1.0 (projected); and Rakudo version 1.0, a partial implementation of Perl version 6 based on the Parrot Virtual Machine.

**Bigfoot spotted with Elvis on passing UFO**

The big story in 2010 is the portable Web. It’s wherever you go, on whatever device you carry.

For developers, the portable Web will be at once familiar and strange. New applications can expect to have a number of new requirements:

- Your Web application must work well on a number of devices.
- Your application must interoperate with the browser and perhaps even work offline.
- With the advent of HTML 5, JavaScript is more important than ever, because each HTML 5 API is based on the language. (If you have yet to master JavaScript coding, yesterday is a good time to start.) Fortunately, HTML 5 works well in modern browsers, as does the JavaScript language, and JavaScript frameworks such as jQuery and Prototype tend to hide browser inconsistencies with aplomb.
- Your back-end code—be it Ruby on Rails 3, PHP 6, Rakudo, or something else—must support a variety of front ends, potentially with wildly different interfaces.
- Your database must scale. What type of database is right? It depends.
Few things change as quickly as technology, and Web technology seems to change faster still. These are heady times for software developers, when a few people in a garage can create the next killer application. Who knows? Maybe you'll make the next headline. See you in the tabloids.

© Copyright IBM Corporation 2010
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)