

Sigma Σ

noblis.

Volume 11 Number 1
October 2011

INSIDE THE DIGITAL ECOSYSTEM

Capitalizing on Connectivity





The spark of innovation
begins with collaboration

At Noblis we know that collaboration can spark a great idea. And in today's connected world, those ideas can come at any time and arrive through many mediums.

Noblis uses science and advanced technology to turn ideas into innovation. And, through our technology-driven, forward-thinking approach, we are delivering next-generation solutions today.

Collaborate. Innovate. Deliver.

noblis[®]
For the best of reasons

**H. Gilbert Miller**

Corporate Vice President
and Chief Technology Officer
Noblis

Connectivity and the User Experience

As I look around meeting rooms, academic institutions, libraries, coffee shops, streets, and even my home, I realize that we have become an “always-on” society. Our reliance on paper and even face-to-face conversations is diminishing. Increasingly, service providers that used to rely on word of mouth from satisfied customers now rely on Facebook, Twitter, LinkedIn, and a host of social media outlets. More customer service lines are urging callers to use their computing devices for interaction.

But it is no longer enough to be simply connected, to be able to say “visit our website.” The website must deliver the kind of user experience that ensures some favorable feedback, whether a five-star Amazon review, or adherence to a desired policy or behavior, or the gratification of being able to view desktop graphics on a smartphone screen. The pressure is on service providers to reach the millions of connected users who will settle for nothing less than a content-rich site that delivers what the provider promises. Major retailers like Walmart and Target have discovered the devastating consequences of not following through on service promises. When the companies had to cancel or delay eagerly anticipated orders, users were dismayed, frustrated, and vocal. This kind of bad user experience can thwart any attempt to keep old customers and attract new ones.

For the federal government workforce, the pairing of connectivity and a rich user experience means providing content and services that engage users and motivate them to complete and possibly build on the agency’s mission. As users become more fully engaged, they are apt to grow their involvement, contributing in unexpected ways that could ultimately decrease operational overhead. In disaster response, for example, the cost of harnessing social media is negligible relative to the resources to enhance formal first-response support.

Conceptually, the goal of retaining users seems straightforward enough, but the range of devices and platforms can be daunting to any organization attempting to understand and enter the mobile universe. This issue of *Sigma* aims to map out a starting point and serve as a guide in quickly implementing design principles that address all user devices—not just desktops and smartphones, but the growing variety of devices in between, laptops, PDAs, and tablets. Our authors give practical advice and design strategies for reaching all segments of user population, from motivating users to stay on your website to understanding how HTML5 and the emerging semantic web will change user experience design.

Ubiquitous computing has been a reality for some time, but organizations are still struggling to understand the new era of service-oriented ubiquitous computing—a world in which the user experience dominates design. As our insight and comfort level with new technology increases, however, struggles and frustration should give way to enthusiasm mixed with a little awe at what these new design tools and strategies can produce. ■



Editor-in-Chief
H. Gilbert Miller, PhD

Digital Ecosystem Edition Editor
Mile K. Corrigan

Editorial Staff
Nancy Talbert
S. Denise Murphy

Production Staff
Carolina Cabanillas
Jennie Doran
Hannah Oh, cover design

Sigma is a publication of Noblis



Noblis: We are the impartial people who use the best of science and technology for the best of reasons—to do what's right and what works for our clients and for the public good.

Noblis publishes *Sigma* to further its work in the public interest. *Sigma* seeks to inform public interest decision makers of recent advancements and accomplishments in a wide variety of scientific and technological areas, including information technology, networking, and life and chemical sciences. Noblis authors base the *Sigma* articles on experiences and knowledge gained from our public interest client work program and our Noblis Research program.

For more information about Noblis, please visit our website, noblis.org, or call Corporate Communications at 703-610-2002.

Volume 11 Number 1
October 2011

Approved for public release, distribution unlimited
3150 Fairview Park Drive South
Falls Church, VA 22042

Copyright © 2011
Noblis, Inc., All Rights Reserved





Features

4 **Toward a User-Centric Digital Ecosystem**

Mile K. Corrigan

As digital platforms become an extension of the user, organizations must look at all interaction channels in terms of the total user experience. They must reshape their digital strategy, whether that is to develop web applications, manage and tailor content, or launch new services.

13 **Responsive Web Design: Enriching the User Experience**

Brett S. Gardner

New web standards are making responsive design practical, allowing designers to create a single website that can adapt layout and content to viewing contexts across a spectrum of digital devices. The result should be a more satisfying experience for any user.

20 **The What, Why, and How of Mobile Applications**

Daniel Y. Na

Evolving a mobile business strategy demands a multifaceted perspective. Not only must organizations identify effective processes, but they must understand and capitalize on the methods and technologies that will shape mobile development's future.

27 **Bridging the Mobile App Gap**

Adam M. Christ

Mobile application development used to mean writing code for each device platform, but emerging web technologies are spawning hybrid development approaches, which promise to change that fragmented picture.

33 **Giving Data a New Face**

Seth A. Blanchard

Excel files and database records will remain the data infrastructure's backbone, but they do a poor job of transforming data into information. Data visualization makes it easier to see patterns, and Internet delivery can promote exchanges that lead to innovation.

39 **No More Black-box Disaster Response**

Matthew B. Holtry and Judy A. Vo

Social networks lay the foundation for "crowd sourcing," which leverages the tweets, texts, and messages that people post during and after a disaster. The challenge is to channel that information through effective website design so that first responders continue to get accurate and timely data.

45 **In Depth: HTML5: What's Different for User Experience Design and the Web?**

Daniel Y. Na and Brian C. DeRocher

HTML5 fixes many of HTML4's problems, emphasizing usability while capitalizing on Internet growth. Organizations that understand its features can better position themselves in the upcoming shift to a semantic web.

Departments

26 ***Sigma*—In Addition**

More from Noblis authors

52 ***Sigma* Spotlight—Fusing Business Process Management and the Social Web**

David W. Vera

The federal government is facing challenges of vast complexity. A cohesive blend of BPM and mobile and social technologies can provide a mechanism to address those challenges.



Toward a User-Centric Digital Ecosystem

Mile K. Corrigan

As digital platforms become an extension of the user, organizations must look at all interaction channels in terms of the total user experience. They must reshape their digital strategy, whether that is to develop web applications, manage and tailor content, or launch new services.



With the advent of Web 2.0, digital systems rapidly became collaborative, bringing a social layer to the Web. Designers focused on how best to unite ideas, promote mobility, and enable multiple devices to access a range of services.

But collaboration and socialization was only a milestone on the journey to the complete integration of physical and cyber space. The economy and prevalence of technology such as wearable sensors and miniaturized cameras are redefining user-centric computing. Although users were certainly a central consideration in traditional system design, pervasive

Inside Track

- Successful organizations will be those that can leverage interaction channels not only to engage users but to alter their behavior.
- Once availability and basic support needs are satisfied, users move to an experience that gives them significance, such as connection, growth, or personal contribution.
- The gaming world has learned much about motivating and influencing users. Gaming dynamics influence the user, while gaming aesthetics makes the experience fun and engaging. Both should be part of any total user experience.
- Many business leaders can't see how to integrate user-centered design into their existing processes. Four simple guidelines can be useful first steps.

computing is taking computing away from the office and living room and into everyday objects. It is the age of the “app” and the smartphone. Through natural interfaces, systems adapt to users, not vice versa. Wearable sensors are feeding user information to medical networks, Facebook and Twitter are collecting information on everything from events to personal tastes and behaviors. Global positioning systems (GPSs) are appearing in buses and taxis, providing unprecedented volumes of location data. Advertising is moving from billboards to interactive, context-sensitive displays that are changing the way entire communities behave.

Fast-forward a few years and there will be no need to deliberately connect to the Web or application platform. All manner of devices will follow a user through a typical day, transparently connecting at just the right times to enhance the journey—whether that’s eating, working, playing, traveling, or even sleeping. GPSs, activity monitors, cameras, and mobile phones will seamlessly collaborate in providing data about user behavior, buying preferences, social patterns, and work habits, adjusting room temperatures, posting energy consumption data or carbon footprints, or issuing reminders to stop and pick up a roast for dinner.

In this digital ecosystem, the user is the center of networked interaction channels, and successful organizations will leverage those channels, not only to engage users but to alter their behavior. The Web is just one interaction channel in this ecosystem. The controversial article, “The Web is Dead, Long Live the Internet,” which appeared in the August 2010 issue of *Wired*,¹ describes a shift from the wide-open web to semiclosed mobile application platforms. The authors argued that consumers are choosing these platforms, not because they’re rejecting the Web but because they often work better. It’s just easier when the screen comes to the consumers instead of the consumers’ having to go to the screen. In March 2011, the *Washington Post* countered the *Wired* article by arguing that web design will catch up to the application market.²

Either way, digital platforms are transforming into a series of personal companions that are as much an extension of the user as they are a data processing machine. To build a cohesive digital brand that truly connects with and influences users, designers and developers must apply human-centered design to reshape their digital strategy, whether that strategy is to develop web applications, manage and tailor content, or launch new services. In the face of the 300,000-plus applications in the App Store, businesses need differentiators to deliver consistent brand recognition and compelling user experiences.

Figure 1 shows the formidable selection of channels in a digital ecosystem, each of which can be exploited to engage and influence users.

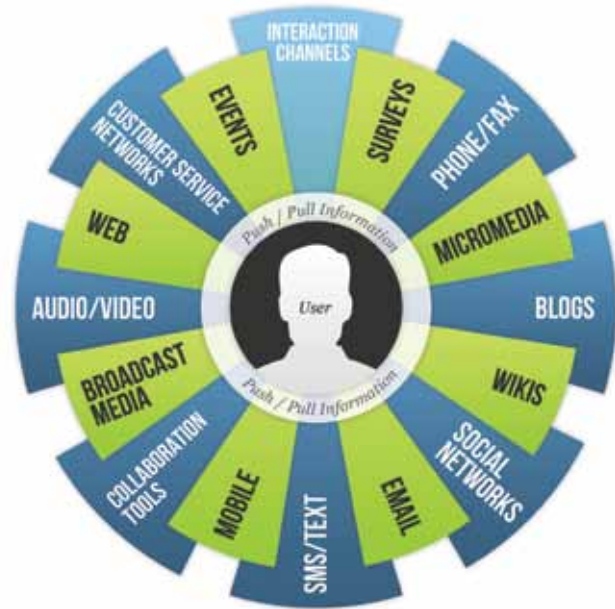


Figure 1. Ecosystem of user interaction channels. Within this digital ecosystem, interaction channels are highly integrated, working together to provide a total user experience.

Understanding the user

The human brain is complex, with both conscious and subconscious thoughts influencing our experiences. Although most people think they make decisions based on conscious thought, in large part subconscious processing governs most decision-making and behavior. Tapping into the emotional midbrain to understand desires, motivations, and contexts for action lays the foundation for understanding how to drive a user’s behavior. Maslow’s hierarchy of needs advocates that, once a person’s lower level needs (such as food and water) are met, that person moves toward self-actualization.³

Some argue that this hierarchy is analogous to the user experience. Once availability and basic support needs are satisfied, users move to an experience that gives them significance, such as connection, growth, or personal contribution. Interaction platforms that can successfully address user needs and emotions promote a viral and addictive effect among their user community, resulting in higher user satisfaction rates and increased brand loyalty. For example, Whitehouse.gov not only provides users with the latest Presidential news and speeches but also enables users to connect more frequently through Facebook, YouTube, Twitter, Flickr, MySpace, Vimeo, iTunes, and LinkedIn. Users can sign up for email alerts and track a wide variety of political issues, including civil rights, immigration, healthcare, family, education, and national defense. The White House Blog

provides a direct line of communication with constituents, providing links to additional resources and a mechanism to share and bookmark popular blog posts.

User research is essential to help organizations surface unfamiliar user needs, perspectives, and goals to provide customers with an enriched experience. Organizations such as Human Factors International have developed training programs geared to the science of persuasion, emotion, and trust to help businesses create successful online interactions that motivate and influence users. The training program provides research-based methods to help organizations understand the human factors that lead to customer conversion. Such understanding then becomes the basis for developing persuasive strategies and forging strong customer relationships.

Many industries are already applying user-centered and persuasion-based approaches, both online and off. As input to its wearable coaching device, Phillips uses human coaches to determine which arguments persuade an individual to adopt a healthy diet and exercise regularly. According to another *Wired* article,⁴ persuasion profiling is just around the corner, and "... it doesn't just find content you might enjoy. It figures out how you think." To test the effectiveness of various persuasive techniques, Dean Eckles, a doctoral student at Stanford University, established an experimental online bookstore that encouraged customers to browse titles and flag those they would buy. Eckles tried several sales tactics, from appeal-to-authority ("Malcolm Gladwell says you'll like this") to social proof ("All your friends on Facebook are buying this book") and tracked which one worked best.⁴ Using process of elimination, researchers were able to increase the effectiveness of persuasive styles on particular individuals by 30 to 40 percent.

Total experience design

Given that understanding the user is important in the new digital ecosystem, how do organizations go about acquiring such insight? One common misstep is to collect user feedback and implement changes accordingly but without first analyzing user motivations. Feedback gathered in this manner is simply not fine-tuned enough to ensure a satisfying user experience. A March 2011 *UX Magazine* article⁵ describes the importance of a razor-sharp focus to truly understand the problem and the users, to analyze behavior, and to translate needs into a creative solution. It offered this advice to organizations:

"Everyone involved in the development of a product or service—including designers, developers, business leaders, and project managers—needs to have the same focus on the end user and understand that ultimately their decisions will affect the user's experience and perception."



Figure 2. Total user experience. Users have additional touchpoints before and after actual interaction. Understanding the full range of user motivations and reflective emotions can help an organization design a more meaningful user experience.

By analyzing users' business *and* emotional sides, organizations can better understand customer demographics, buying patterns, feelings, and triggers. The response to such insight is fostering emergent fields, most prominently, cross-channel design, or *total experience design* (TxD). TxD views the user experience as transcending a specific medium, delivery mechanism, or interaction channel.

Figure 2 depicts the user experience as having three parts, not just end-to-end interaction, but the motivations before and the reflective emotions and behaviors after. This in-depth user experience is the essence of TxD: It is the consideration of an overall potential customer experience, regardless of medium or other boundaries. Its aim is to draw out what elements must coordinate or orchestrate to create "a memorable, delightful, valuable experience that people want to talk about."⁶

The before, during, and after in the user experience are *touchpoints*—contact points that a user experiences when interfacing with any interaction channel, whether print, web, broadcast media, or product packaging. By analyzing these touchpoints, new and sometimes unexpected insights emerge to inform the enhancement of customer experiences. Touchpoints can be static, interactive or human. The *static* touchpoint is the recipient's emotional response to a channel's packaging, such as a Netflix subscriber's excitement at seeing the company's trademark red envelope containing the latest DVD. *Interactive* touchpoints involve two-way communication. As the recipient adds DVD titles to her queue, the system updates the queue in response to that action. The process of adding and updating is an interactive touchpoint. *Human* touchpoints involve the user and one or more others, such as calling technical support or customer service for assistance.

Touchpoint analysis typically involves observing user behavior, identifying the use context, strictly attending to detail, looking

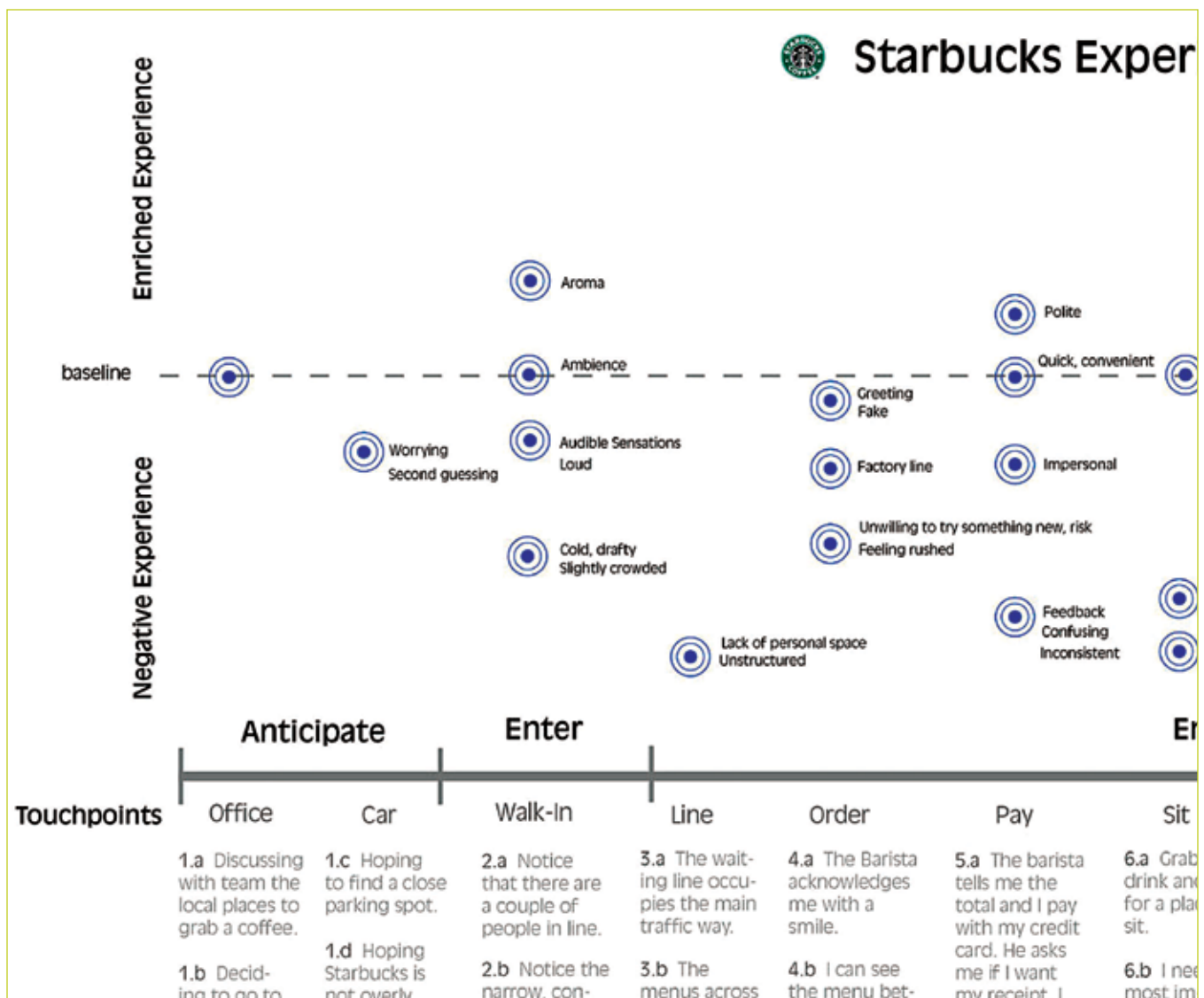


Figure 3. Excerpt of a Starbucks user experience map. The map reflects customer feedback and attitudes at particular touchpoints in a typical customer journey from office to order. (Image courtesy of Eric Berkman, Little Springs Design.)

for user workarounds, and examining related business processes and user attributes. The user experience map, a popular method for analyzing user touchpoints, breaks down a user's experience by plotting individual touchpoints and satisfaction across each interaction mode during the engagement.

Because a user experience map thoroughly assesses a user's emotional response throughout the process or interaction, it is an excellent mechanism for identifying opportunities to optimize processes and improve user experience. In Figure 3, for example, the user experience map clearly shows that the customer's poor experience is due to the store design, not the product quality. Although this example comes from the profit world, this type of customer analysis is extremely useful when modeling user experience across multiple interaction channels, such as the Web, help desk, kiosks, and email. As the federal

government continues in its mission to improve service delivery and accessibility to information through various channels—the Web, mobile services, and public data exchange—multimodal touchpoint analysis will be essential in understanding the total citizen experience.

Interaction framework

The results of examining user touchpoints form the basis for developing an interaction framework, which the organization can use to develop key user scenarios, early models, and prototypes. User analysis has identified preferences and behaviors, enabling the user experience team to understand where their users want to connect and what platform attributes they desire. Table 1 shows some of the tools that designers can apply in developing attributes of the user experience. Storyboarding is a common

method for describing system functionality and use. It is similar to use cases or modeling in the Unified Modeling Language, but it also offers a visual representation of the interaction, specifying key user interface elements and establishing a common language among concept designers, developers, information architects, engineers, and business stakeholders.

The defined interaction framework enables user experience teams to develop detailed sketches, mock-ups and wireframes that lead to more detailed design. For web-based projects, teams might implement early prototypes or functional mockups to examine behavior at the beginning of the process and iterate rapidly to produce more refined development specifications. Designers and developers tend to avoid redesign and refactoring once development is underway, which means that the sketching and storyboarding process is critical. With any framework design approach, frequent communication and feedback is integral to delivering an on-target solution.

Reuse-based design

Because so many platforms are available, organizations must design for reuse from the start. Designing for reuse means ensuring that content and functionality serve more than one purpose and are optimized for each platform's unique characteristics. By leveraging delivery channels, organizations can create microexperiences through social media outlets, blogs, mobile applications, RSS feeds, and YouTube. Social media lets organizations syndicate tools, content, and communication to connect more intimately and more frequently with users.

An example is the General Services Administration's (GSA's) Sustainable Facilities (SF) Tool, which the GSA Office of Federal High Performance Green Buildings launched and Noblis, CTG Energetics, and the U.S. Green Buildings Council implemented. This highly interactive, web-based framework consists of a custom content-management system that syndicates live

sustainable design content to its sister application, SF Mobile. SF Mobile is a cross-platform mobile application for iPhone, Android, and BlackBerry® devices that has been optimized for characteristics of the mobile user-interaction channel—simplicity, immediacy, and portability. Both the website and mobile application offer a consistent user experience by providing access to sustainable strategies and design guidance for interior space projects. Figure 4 shows screenshots of both the SF website and the SF Mobile application, each tailored for its particular platform while serving the same content.

Game strategy to influence behavior

Although hardly a new concept, game strategy has become highly relevant in the digital ecosystem's evolution. Game play must be both content- and context-specific, which requires deeply understanding the user community. The Mechanics, Dynamics, and Aesthetics (MDA) framework,⁷ for example, formalizes game consumption and breaks games into three components to establish design counterparts:

- **Mechanics.** Particular game components at the level of data representation and algorithms,
- **Dynamics.** Runtime behavior of the mechanics acting on player inputs and each other's outputs over time, and
- **Aesthetics.** Desirable emotional responses evoked in the player when he or she interacts with the system.

Of these, game dynamics and aesthetics are the most relevant to total user experience design.

In his July 2010 “The Game Layer on Top of the World” presentation,⁸ Seth Priebatsch described the move to shape user behavior through game dynamics, which has led to “gamification”—the integration of game mechanics into nongame

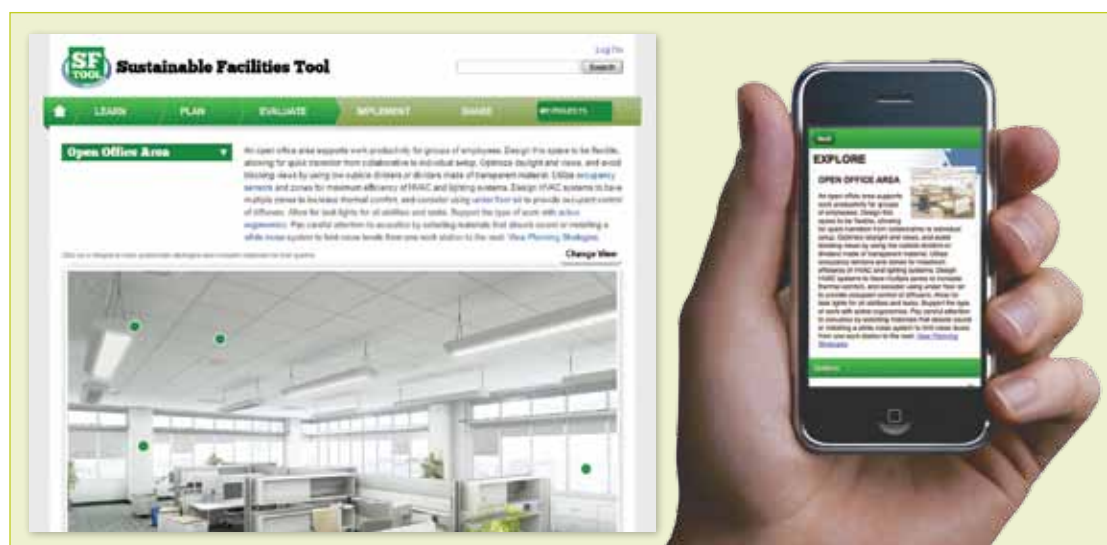


Figure 4. The General Services Administration Sustainable Facilities (SF) Tool. The SF website and SF Mobile application offer the same content, but tailored for their respective platforms. The tool is an example of how to design for reuse.

Table 1. The user experience creative toolbox.

Tool	Design Stage	Purpose	When to Use
Mind Map 	Brainstorming	Builds a system of thoughts around a starting point	To visually elicit thoughts and connections
Personas 	User Research	Represents a user group's goals and behavior	To define users' social and demographic characteristics, including desires, habits, and cultural considerations (after real user observation)
Customer Journey Map 	User Research	Describes a user's journey and represents touchpoints that characterize interaction	To synthesize user touchpoints and describe information and physical platforms involved
User Experience Map 	User Research	Assesses the user's emotional response throughout a process or interaction across every touchpoint	To measure user satisfaction across each interaction mode during the user engagement
Affinity Diagram 	Analysis/Modeling	Organizes and relates ideas and user data	To expose design principles from the collection of user observation data following brainstorming and user research
Storyboard 	Analysis/Modeling	Represents user scenarios through sketches or images in a narrative sequence	To describe each user scenario with realism in narrative form, exposing key aspects of the user experience
Moodboard 	Early Design/Prototyping	Composes colors, pictures, and materials to evoke style and overall user feel	To communicate design concepts and visually illustrate the atmosphere surrounding a product or service
Wireframe/Mockup 	Early Design/Prototyping	Models the user experience by illustrating key design elements and features	To visually depict layout or arrangement of content related to user scenarios including interface elements, navigation, and functionality
Requirements 	Requirements Specification	Provides requirements for the implementation team, ensuring alignment of features and functionality to user goals, needs, and preferences	To help developers understand how the system or application will be used and to validate quality assurance and quality control processes during testing
Prototype 	Prototyping/Detailed Design	Simulates the user experience	To test the solution through active user and stakeholder participation

Common Game Dynamics

Organizations can choose from nearly 50 game dynamics to influence user behavior. These are the most common ones:

- **Achievement.** The user receives a virtual representation of an accomplishment such as a badge, a level, a reward, or points which drives continued user interaction.
- **Avoidance.** Opposite of the achievement dynamic, avoidance induces user behavior by instituting a punishment. Survival games use this dynamic by punishing users with draining health and happiness scores associated with lack of activity.
- **Appointment.** To succeed, the user must return at a predefined time to take a predetermined action. Happy Hour and Farmville are examples.
- **Countdown.** Users are given a certain amount of time to complete an activity which increases user activity until time runs out. Several shopping websites such as HauteLook, Zulily, and Gilt Groupe rely on this game dynamic to encourage users to buy quickly before items are removed from their timed shopping cart.
- **Influence and status.** The ability of one player to modify the behavior of another's actions through social pressure. Credit card color, "Modern Warfare" (game), and a report card are examples.
- **Progression.** Success is granularly displayed and measured through the process of completing itemized tasks. LinkedIn status and "World of Warcraft" are examples.
- **Communal discovery.** An entire community is rallied to work together to solve a challenge. The Digg Leaderboard and the Defense Advanced Research Projects Agency's Network Challenge are examples.

environments to increase audience engagement, loyalty, and fun (<http://gamification.org/wiki/Encyclopedia>). Priebatsch claims that gamification with only seven game dynamics can get anyone to do anything, and his SCVNGR gaming firm lists 47 game dynamics in its Secret Game Mechanics Playdeck (<http://techcrunch.com/2010/08/25/scvngr-game-mechanics/>).

Most popular websites implement one or more gaming strategies. Farmville uses the appointment dynamic, requiring the user to return at a specific time and conduct a task to receive a reward. LinkedIn uses progression dynamics to influence users to populate their user profiles with more personal information, which it can then use to offer the user information-exchange services for a fee. The profile displays the user's "success" measured through the accumulation of profile attributes. The sidebar "Common Game Dynamics" describes other strategies for influencing users.

Game aesthetics is what makes the experience fun for the user. Aesthetics can take any of eight forms, from sensation (game as sensual pleasure) to discovery (game as uncharted territory). By considering user-driven aesthetic goals for design, MDA researchers could better understand how to optimize game dynamics on the basis of emotions such as fear, anger, pride, sadness, joy, and excitement. These emotions trigger natural

instincts such as survival, competition, or curiosity, effectively pushing the player to act on the emotion. "Angry Birds," the best selling paid application as of June 2011, mixes humor and aggression to trigger the instincts of competition, curiosity, and revenge, prompting the user to hurl angry birds at pigs. According to Tomas Chamorro-Premuzic, a psychologist from the University of London, "Angry Birds" gives people a release for their aggressive impulses. "They might not realize it or thank you for pointing it out, but the game has a cathartic effect, letting them act on their aggressive tendencies."⁹

Evolve-and-adapt strategy

Continuously measuring user experience is critical to an effective digital strategy. User feedback should be at the center of designing an integrated communication platform that will expose user desires, needs, and preferences as the organization's understanding of the user experience evolves.

Customer-focused organizations, including eBay, Amazon, and Apple provide multichannel feedback loops to improve their products and services in their users' eyes. They also invest heavily in user experience to build their brand and increase customer satisfaction. Such strategies have proven commercial success: In a 2009 Forrester survey that asked consumers to rate their website experience, these organizations dominated the top ten.¹⁰ Indeed, the focus on user experience has intensified to the point that companies exist solely to help compile, analyze, and deliver user feedback, including UserVoice, IdeaScale, YouSuggest, CrowdSound, and Get Satisfaction.

Federal agencies have also implemented feedback loops to evolve key programs and initiatives. The GSA's Carbon Footprint Tool is designed specifically to assist federal agencies in managing their greenhouse gas emissions to comply with Executive Order 13514. The tool automatically captures and categorizes user feedback to drive future development. Before launching the tool, the agency conducted a beta test in alignment with the Public Sector Protocol road test. The launch reflected feedback from 55 federal agencies—insight critical to the development of a tool that must serve the breadth of the federal community. The agency not only collected feedback but synthesized it into prioritized user-feature groups that led to the tool's refined development. Indeed, GSA's feedback-driven approach earned the GSA Carbon Footprint and Green Procurement Tool the *Achievement in Inauguration* award from the Center for Environmental Innovation and Leadership.

Clearing the obstacles

With so many benefits to the total user experience design, why don't more organizations use it? One reason is that many business leaders fail to grasp how user-centric design can help them achieve their business goals and objectives or how they

can integrate such design practices into their existing processes. Late user interface design, conflicting user feedback, and the incorrect measurement of user analytics can easily derail a potentially successful user experience project.

By taking four relatively simple steps, organizations can begin to reap the rewards of user experience design:

- **Collaborate.** Developers and designers must collaborate to save time and reduce costs.
- **Take more upfront time.** Spending more design time up front ensures frequent communication throughout all project phases.
- **Prototype before development.** Prototypes enable course corrections before costly infrastructure is in place.
- **Align metrics to success factors.** Develop metrics that align to critical success factors, such as saving customer's time.

Developers often misunderstand this last obstacle-clearing strategy. They can measure “saving time” by the time a user takes to complete a task and the number of clicks. They can measure “reduced technical support” by the number of help desk calls, and they can measure “increased user satisfaction scores” by user survey results. Performance-based metrics, such as task completion time, are just as important to measure and evaluate as the more familiar perception-based metrics, such as user satisfaction and ease of use. Comprehensive measurements of the user experience—involving both qualitative and quantitative methods—are essential in analyzing and responding to user results.

Next steps

Clearly, success in the future digital world won't be based solely on the user experience. It will require an integrated digital strategy that incorporates human-centered design philosophy, good business sense, and the consideration of both customer voice and business objectives. Forrester defines digital customer experience strategy as

“A plan that guides the activities and resource allocation needed to deliver experiences that meet or exceed customers' expectations within and across digital interaction points.”¹¹

By capitalizing on psychology, business and end-user empathy, continuous feedback cycles, and communication, organizations will have the power to strategically influence and drive customer behavior.

As Human Factors International noted, user experience is transcending tactics at the enterprise level. As user centricity moves up the value chain, more organizations are approaching

customer experience as a differentiator and brand strategy.¹² To effectively evolve user experience across an enterprise, organizations must take six key steps:

- develop a customer experience strategy that aligns business objectives and user goals;
- involve key stakeholders and secure executive support;
- embrace customer-focused design principles in every organizational unit that connects to the end-user experience, including branding, sales, development, design, customer service, and user support to provide consistency across delivery channels;
- train employees in user experience principles, methods and tools;
- collaborate frequently to ensure smooth touchpoint transitions; and
- measure both successes and failures to develop repeatable tools and processes and improve from lessons learned.

Conducting off-site meetings and targeted user workshops to engage groups that would normally not work together is one way of surfacing opportunities to optimize business processes and make user-focused improvements. Large and widely-dispersed organizations might need to hire a chief experience officer or director of user experience to serve as the user experience champion and evangelist. Having such a position ensures that groups communicate, collaborate, and commit to an integrated, consistent, and positive user experience.

In this issue

This issue of *Sigma* presents several emerging technologies, design patterns, and digital interaction channels that will comprise the future user-centric digital ecosystem. These articles demonstrate how organizations can effectively apply user experience not only to respond to future technology trends and refreshment cycles but also to rapidly evolve with the ever-changing user.

In “Responsive Web Design: Enriching the User Experience,” Brett Gardner describes the future of web design patterns and methodologies that will more responsively adapt to the changing screen size of evolving web-capable digital devices. The article describes the new features of Cascading Style Sheets, version 3 (CSS3) that enable this design methodology and presents a user-centered approach to dynamic web application design and development—whether the device is a smartphone or a high-resolution desktop display.

The next few articles deal with mobile development, which has become big business for many companies. Traditionally the need to accommodate multiple platforms has stymied mobile

applications development, forcing the need to learn different development languages and maintain separate mobile code bases. Standards-compliant mobile browsers are changing that picture, making it easier for companies to build mobile web applications with a single code base to reach users across diverse mobile operating systems. Not only does this circumvent the need to write specific programs for specific devices, it also provides a means of avoiding the increased pressure from companies such as Apple to extract revenue from mobile application downloads.

In “The What, Why, and How of Mobile Applications,” Daniel Na exposes several considerations facing mobile application development including the increased prevalence of mobile devices, browser compliance across mobile devices, and current implementation barriers.

New technical strategies that exploit emerging web technologies can also help companies develop applications across devices, as Adam Christ describes in “Bridging the Mobile App Gap.” Organizations will come away with a better understanding of mobile web-application architecture, strategies to replicate native applications using cross-platform web technologies, web application conversion techniques, and emerging mobile testing tools. More important, they will see how to leverage this understanding to enhance content delivery and increase distribution.

“Giving Data a New Face” complements the previous article by taking a practical look at web toolkits and applications for building interactive data visualizations. Author Seth Blanchard describes the benefits of using such tools and offers strategies for organizations to avoid common pitfalls.

Sigma’s cover features wrap up with a look at how mobile technology and social media can save lives. In “No More Black-box Disaster Response,” authors Matthew Holtry and Judy Vo describe how social convergence theory and user experience design motivate interaction strategies. Crowds are self-organizing and focused on problem-solving in the aftermath of natural and manmade disasters. Through social media, people request help, assess the situation, and even provide triage from thousands of miles away—all at no charge to the responder services. Although this crowd sourcing can enable faster relief, it can just as easily give a picture of the disaster that is distorted, and filled with extraneous details that cripple decision-making. The article describes novel ways to use crowd sourcing and social media to generate useful, actionable data for responders.

Finally, *In Depth* takes a hard look at how HTML5’s features are already addressing problems, such as the need for third-party plug-ins, that have plagued HTML4-based websites. Authors Daniel Na and Brian DeRocher explain the differences between HTML4 and HTML5 and how HTML5’s new features will support the new semantic web. “HTML5: What’s Different for User Experience Design and the Web?” explains in practical

terms how this new language will affect web technology and how it will allow organizations to reach users in innovative ways that were not possible with HTML4.

Customers have high expectations and expect on-demand service and valuable experiences that are functionally, emotionally, and aesthetically satisfying. To retain existing customers and gain new ones, both public- and private-sector organizations must bring compelling experiences to the user. Organizations such as Target, Bank of America, and Nike continue to raise the bar in leveraging technologies to provide high-value, consistent user experiences across multiple platforms—from browser, desktop, kiosk, and console to handheld devices. As the Forrester survey revealed, such strategies have proven their worth. Recognizing that total user experience design translates to market success, these organizations continue to invest heavily in user experience to improve market positioning, build their brand, maintain customer loyalty, and increase customer satisfaction. In this arena of ubiquitous, pervasive computing and rapidly evolving user-centered technologies, it would be wise to follow their model of user-focused design, development, and strategy. ■

References

1. C. Anderson and M. Wolff, “The Web Is Dead. Long Live the Internet,” *Wired*, Aug. 2010; www.wired.com/magazine/2010/08/ff_webrip/all/1.
2. R. Pegoraro, “No, Wired, the Web is Not ‘Dead,’” *The Washington Post*, Aug. 18, 2010; http://voices.washingtonpost.com/fasterforward/2010/08/no_wired_the_web_is_not_dead.html.
3. A.H. Maslow, “A Theory of Human Motivation,” *Psychological Review*, 50, pp. 370–396, 1943.
4. E. Pariser, “Welcome to the Brave New World of Persuasion Profiling,” *Wired*, Apr. 26, 2011; www.wired.com/magazine/2011/04/st_essay_persuasion_profiling/.
5. L. Kitson, “User-Led Does Not Equal User-Centered,” *UX Magazine*, Mar. 17, 2011; <http://uxmag.com/strategy/user-led-does-not-equal-user-centered>.
6. P. Dawson, *A Definition: Total Experience Design*, EMC Consulting Blog, Dec. 1, 2009; <http://consultingblogs.emc.com/pauldawson/archive/2009/12/01/a-definition-total-experience-design.aspx>.
7. R. Hunicke, M. LeBlanc, and R. Zubeck, *MDA: A Formal Approach to Game Design and Game Research*, Northwestern University, 2004; www.cs.northwestern.edu/~hunicke/MDA.pdf.
8. S. Priebatsch, “The Game Layer on Top of the World,” *TEDx Talks*, July 2010; www.ted.com/talks/lang/eng/seth_priebatsch_the_game_layer_on_top_of_the_world.html.
9. J. Svetlik, “Angry Birds: The Psychology of Its Success,” *Electricpig*, Mar. 9, 2011; www.electricpig.co.uk/2011/03/09/angry-birds-the-psychology-of-its-success/.
10. M. Burns, “How Satisfied Are Consumers with Website Experiences in 2010?” *Forrester Research, Inc.*, June 14, 2010; www.forrester.com/rb/Research/how_satisfied_are_consumers_with_web_site/q/56315/t/2.
11. R. Rogowski, “Why You Need a Digital Customer Experience Strategy,” *Forrester Research, Inc.*, Mar. 18, 2011; www.forrester.com/rb/Research/why_you_need_digital_customer_experience_strategy/q/58252/t/2.
12. E. Schaffer and S. Weinschenk, *The Evolving Institutionalization of Usability: User Experience as Strategy*, Human Factors International, May 21, 2009; www.humanfactors.com/downloads/documents/EvolvingInstitutionalization_whitepaper.pdf.



Mile K. Corrigan is a manager at Noblis. Her research interests include user experience design, solution architecture, application development, and mobile systems. She received a BS in management science and information technology from Virginia Polytechnic Institute and State University. Contact her at mile.corrigan@noblis.org.



Responsive Web Design: Enriching the User Experience

Brett S. Gardner

New web standards are making responsive design practical, allowing designers to create a single website that can adapt layout and content to viewing contexts across a spectrum of digital devices. The result should be a more satisfying experience for any user.

As a greater selection of devices become able to access the Internet, website designers are finding it harder to predict user context. Traditional best practice for website design advocates using a minimum target resolution—a kind of **low-end catchall** for the browsing windows on various devices. The idea is to gather statistics on website visitors and their screen resolutions and use the results to identify the lowest resolution that would accommodate some acceptably high number of users. The minimum requirement then dictates important design decisions, from the site's canvas size in Photoshop to the architecture of a fixed-width layout.

Unfortunately, creating fixed-width websites on the basis of a **minimum target resolution** (typically 1024 × 768), presents both short- and long-term problems. In the short term, it **ignores users with more capable hardware, giving them only a limited user experience**. In the long term, it **guarantees a site redesign**

every few years to accommodate changes in minimum target resolution as new devices enter the market.

Building a fixed site to minimum requirements can also doom some mobile phone users to tedious panning and zooming. As “The What, Why, and How of Mobile Applications on p. 20 describes, mobile devices are the fastest growing segment of the web access market, with smartphone sales already outpacing computer sales. Businesses needing to address this burgeoning market segment are forced to create additional websites for smartphone users. The rub is that each device has a different form, and as device variety expands—feature phones, smartphones, and tablets—the cost of building and maintaining a website for every mobile device type as well as for desktop display variations like widescreens becomes prohibitive.

The good news is that website developers now have new web standards like Hypertext Markup Language, version 5 (**HTML5**), and Cascading Style Sheets, version 3 (**CSS3**), enabling them to **design and build user-sensitive sites that respond to a range of contexts and device capabilities**.

From One Web to responsive design

In 2005, the World Wide Web Consortium (W3C), an international community dedicated to the development of web standards, recognized that mobile device variety would slow mobile web growth. In “Scope of Mobile Best Practices” it noted the challenge facing website developers:¹

“To cope with highly differentiated capabilities and limitations of mobile devices, content authors and service developers are often forced to deploy multiple versions of their offerings and/or rely on widespread use of adaptation techniques.”

Inside Track

- Developing fixed-width websites based on a minimum target resolution limits user experience, leading to increased costs to develop and maintain separate targeted sites for emerging digital devices.
- With older standards, the only options were to optimize design for some users or create a website for each device type. New web standards, including media queries, are offering new strategies.
- Using responsive web design, businesses can create websites capable of adapting layout, content, and appearance to optimize user experience across devices of varying sizes and capabilities including smartphones, tablets, and widescreen computers.
- Organizations that want to implement responsive web design into their current practices can take several immediate first steps, such as implementing minimalist or mobile-first design.

Looking at a solution as well, the W3C went on to describe their vision of One Web, a “seamlessly integrated Internet” that would have “content accessible with equal ease” to desktop and mobile device users.¹

In 2008, the W3C revisited its One Web vision, underlining the need to make the same information and services available to users regardless of device.²

This work toward a seamless Internet has been foundational to the creation of web standards like HTML5 and CSS3, which can detect and respond to digital device capabilities. Concurrent with the development of these standards was a call to implement *responsive web design*. In a seminal article with the same title,³ Ethan Marcotte explained that **responsive web design aims to combine HTML5 and CSS3 capabilities with a new design methodology for a website architecture that would adapt to browsers of any size**. Essentially, responsive web design was the first methodology proposed to help realize the One Web vision.

Elements of a responsive design

In a blog entry,⁴ Marcotte outlined a method for creating fluid layouts that are screen-resolution agnostic and “future proof,” capable of dynamically changing according to user context. He described responsive design as having three parts:

- a **fluid layout that uses a flexible grid**, which in turn ensures that a website can scale to a browser’s full width;
- **images that work in a flexible context**, whether fluid themselves or perhaps controlled through overflow mechanisms; and

- **media queries**, which optimize the design for different viewing contexts and spot-fix bugs that occur at different resolution ranges.

With a fluid layout, flexible content, and web standards that can sense display capabilities—size, resolution, pixel density, and orientation—designers can finally create context-sensitive, adaptable websites. The result should be an improved experience for all users that access the site, not just those in a minimum target range.

Fluid layout

To build a website with a fluid layout, **the designer specifies content area width as browser window percentages**. In a fixed layout, the width is static and given in pixels; it does not adjust to the user’s screen size, which leaves empty spaces in high resolution displays. Fluid layouts are dynamic and user-sensitive—**adapting to the available real estate on the user interface and providing increased content accessibility**.

A popular website development practice is to use a grid system to produce layouts. Grid systems draw on a set of base styles for columns and gutters to streamline the development of website layouts with user-friendly content alignment and spacing. Although currently popular grid systems are primarily for fixed layouts, grid systems for fluid layouts are gaining ground. Open-source options for fluid-grids include the 1140 Grid (<http://cssgrid.net/>) and the Fluid 960 Grid System (www.designinfluences.com/fluid960gs/).

Flexible media

Layouts based on percentages resize gracefully according to the size of the browser window rendering them. However, it is problematic to ensure that the content within a site resizes

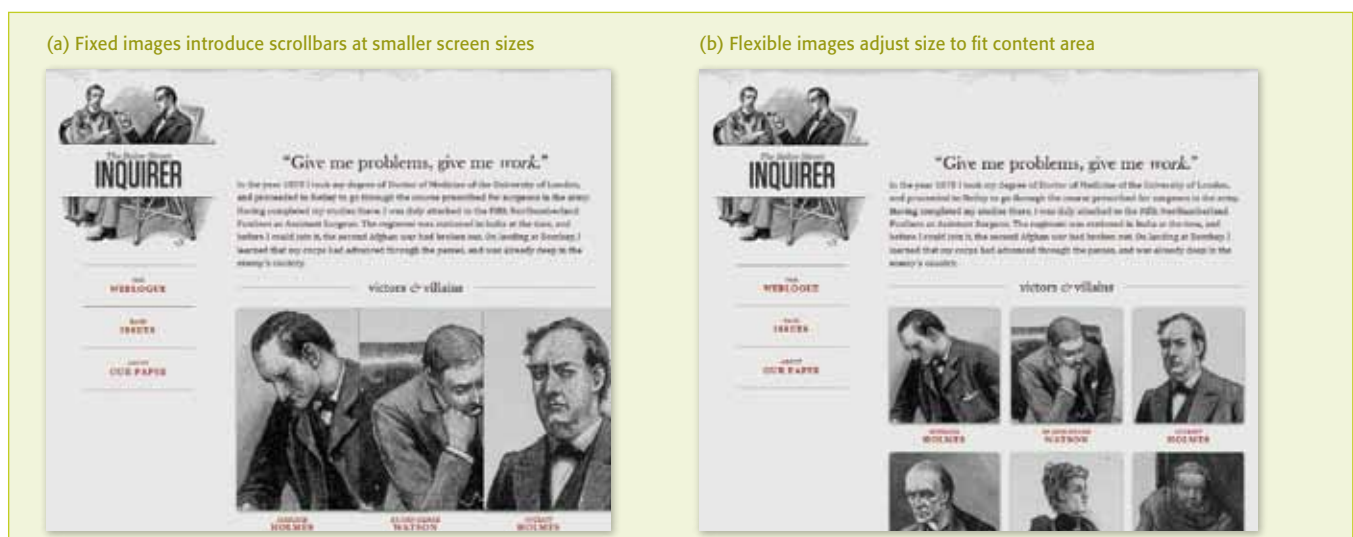


Figure 1. Scaling content and images concurrently. In (a), the images dictate content width, maintaining scale regardless of their container size. In (b), which uses a fluid layout and the max-width property, the images scale according to their container. (Image from Ethan Marcotte; www.alistapart.com/d/responsive-web-design/ex/ex-site-flexible.html.)

concurrently. Although text will wrap according to its parent container's width, media objects (images or videos, for example) have a set default size. Placing a media object in fluid-width content will prevent that area from ever scaling below the media object's width. In Figure 1a, for example, the three character images are only partially present because the media objects (in this case, the pictures) are disrupting the site's fluid-width behavior and introducing scrollbars.

CSS addresses this problem with its **max-width property**. Scaling a media object according to its parent container is as simple as using this statement, which **ensures that the browser will scale the object to its container size if the container's width falls below the object's original size**:

```
img, object {max-width: 100%}
```

Figure 1b shows the same site with the max-width property applied, causing the images to scale according to their content area's fluid width.

Modern browsers, such as Internet Explorer 8+, Firefox 3+, Chrome, and Safari fully support the max-width property. For older IE and Firefox versions, website designers must use a browser-specific stylesheet instead. Such legacy browsers can be quirky when rescaling certain media types. If a website must support these browsers, the designer can also incorporate a JavaScript solution to handle media scaling on the browser's behalf.⁵

Media queries

Regrettably, fluid layouts come with certain usability problems. Figure 2 shows a two column blog layout. The left column, designed for side navigation, has a width of 20 percent, which is fine for a display of 1024px in Figure 2a. However, in a smartphone window, Figure 2b, which is approximately 320px,

the column width is 64px. With the 12-point font necessary for readability, the column would have at most two words per line, which would severely degrade the user's reading experience.

Yet another readability problem arises when the user has a widescreen monitor, which has a typical resolution of 1920x1080. Maximizing the browsing window on such a monitor increases the Content column to a width of 1536px. At that width, line length far exceeds the ideal 12 words per line, increasing the distance between the end of one line and the beginning of the next. Once again, **the reading experience is suboptimal**. This time, the reader will find it hard to track sentences and must hunt for the beginning of the next line.⁶

CSS3's media queries directly address these usability problems by **allowing browsers to serve different styles for different viewing contexts**. CSS2.1 introduced the ability to specify a stylesheet's media type, including screen, print, and handheld. CSS3 greatly expands support for media queries, adding the ability to target media features such as screen and device width and orientation.

The following media queries accommodate styles common to smartphone forms with device resolutions up to 480x854, such as the iPhone and Android phones.

```
@media only screen and (min-device-width:320px) and (max-
device-width:569px) {
    /*Styles for Smartphones (portrait and landscape)*/
}
@media only screen and (min-width:321px) and (max-device-
width:569px) {
    /*Styles for Smartphones (landscape only)*/
}
```

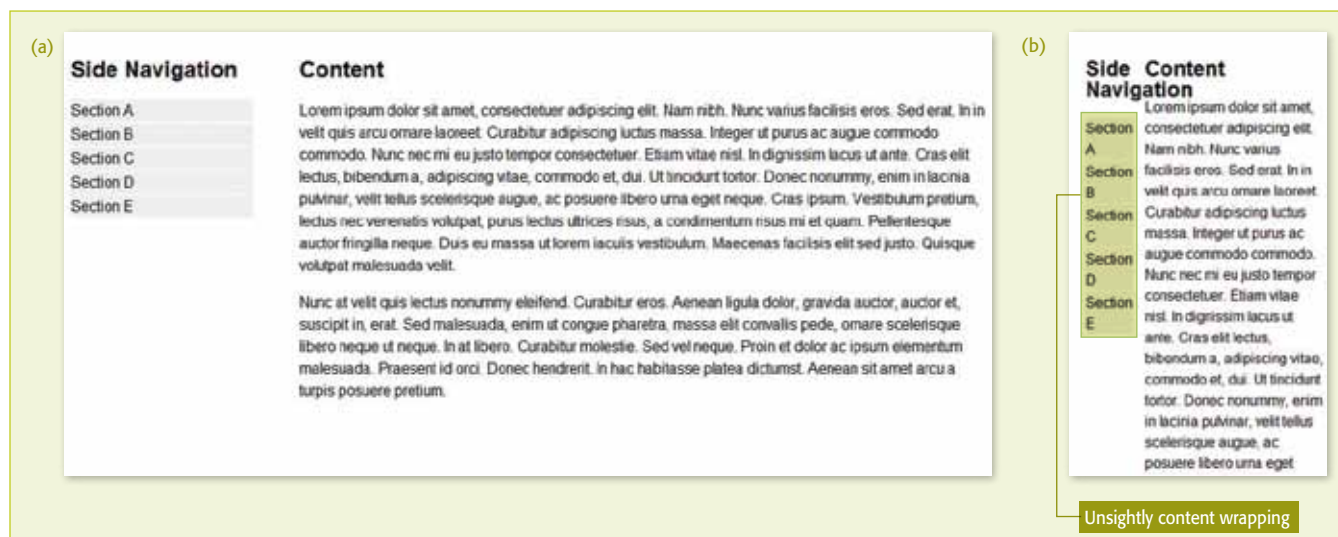


Figure 2. The problem with scaling a fluid layout. (a) At 1024px, content wrapping in the Side Navigation column is acceptable, but at (b) approximately 320px, which is smartphone width, left-column wrapping severely compromises readability.

Table 1. Sampling of devices and their resolution or orientation.

Device Type	Resolution Width or Orientation
Smartphone	Less than 570px
Tablet	Supports orientation
Small-screen laptop and desktop	570px – 1280px
Widescreen monitor	Greater than 1280px

When setting content to 'width=device-width' in the viewport metatag, smartphones with larger screens recalculate device width on a scale to 320. For example, Motorola's DROID 2 with a resolution of 480x854 recalculates its resolution as 320x854/(480/320), translating to 320x569 for evaluation in media queries.⁷ The second media query introduces a min-width constraint that allows styles for the devices' landscape orientation.

The retina display on the iPhone 4, although featuring a high qHD resolution (540x960), interprets media queries at 320x480 with a device-pixel ratio of 2. Designers can use the media query "-webkit-min-device-pixel-ratio: 2" to handle iPhone 4 styles, such as higher resolution images.

Adjustments for resolution. From smartphones to laptops and HDTVs, a device that accesses the site can be inches or feet or anything in between with its own resolution range. Devices such as tablets support orientation changing, providing two possible screen widths. It is not feasible to incorporate all possible resolutions into a website design, but grouping resolution ranges into categories, as in Table 1, can focus design decisions.

By adding media queries for these resolution and orientation ranges, website designers can effectively adjust layout and content for the device context, ensuring that the user has a richer viewing experience.

Adjustments for layout and content. The following media query addresses the problem in Figure 2b, adjusting the fluid two-column layout to a one-column layout for small screen devices.

```
@media only screen and (min-device-width:320px) and (max-device-width:569px) {
  #sidenav {float:left;width:100%;}
  #content {float:left;clear:left;width:100%}
}
```

The min- and max-device-width declarations in this media query target mobile devices such as smartphones with screen resolutions up to 480x854. Modifying the float properties of the side navigation and content areas adjusts the site to a one-column layout. When a user views the site on a smartphone, the side navigation and content appear stacked and expand to the full screen width, maximizing their readability.

To further improve the smartphone user's experience, the designer can use **content substitution**. Even with side navigation above the content, a user must scroll to view page information. The next media query **hides the side navigation and uses dropdown navigation to avoid scrolling**:

```
@media only screen and (min-device-width:320px) and (max-device-width:569px) {
  #sidenav ul {display:none;}
  #sidenav form {display:block;}
}
```

Figure 3 shows the final site layout after using both media queries.



Figure 3. The blog in Figure 2 redesigned using media queries. (a) 1024px and (b) 320px. Media queries let designers adjust not only the layout but also the content, in this case substituting dropdown navigation for the side navigation in Figure 2's layout. The site is now optimal for access by both small screen computers and smartphones.

Optimizing for Speed

Page loading time is an integral part of providing a responsive user experience, and extensive web research suggests that it correlates directly to how long users will stay on a website and how satisfied they are with the interaction, as well as to conversion rates and search engine ranking.

Figure A shows the time users are willing to spend waiting for a web page to load. According to surveys conducted by Akamai and Gomez.com, 40 percent of the users who go to a website will abandon it if page load time exceeds three seconds. And even a one-second delay in page response can reduce conversion by seven percent.¹ For an e-commerce site making \$100,000 per day, for example, a one-second delay could potentially cost \$2.5 million in lost annual sales.

Google also measured how page speed affects user satisfaction. In a 2006 experiment, it adjusted the number of results returned in a search from 10 to 30 for a subset of visitors. According to Marissa Mayer, then Google's vice president of search products and user experience, traffic and revenue from Google searches in the test group dropped 20 percent because page load time increased by half a second for those who needed the additional results.²

Google believes so strongly in the effect of site speed on user satisfaction, that they include it as one of the 200+ factors that determine search rankings.³

With more than half of all new Internet connections coming from mobile devices, ensuring great performance for mobile users has become critical.⁴ Optimizing for speed on mobile has its own challenges, including the need to account for reduced bandwidth and increased latency on mobile networks and reduced processing power on mobile devices.

A few techniques to optimize a site for speed go a long way toward mitigating these issues.

Reduce requests

Each request a page makes to the server to grab an image, script, or Cascading Style Sheets (CSS) file has an associated time cost. This cost increases with mobile devices because of the connection latency inherent in mobile 3G and upcoming 4G networks. Surprisingly, although download speed can be up to 10 times faster, connection latency is identical between 3G and 4G networks.⁵ Indeed, a Blaze study found that request number is the top slowdown factor for mobile sites.⁶

To reduce the requests a page makes, designers can

- combine CSS and JavaScript files,
- embed binary data for images in CSS files via data URIs, and
- load images and content only when scrolled into view.

Reduce page size

It makes sense that the less data transferred to a page, the faster it will download and display. With download speeds on current mobile 3G networks about three times slower than those for cable or DSL, reducing page size can be crucial to decreasing page load time.⁷

To reduce page size, designers can

- use CSS3 instead of images for stylistic elements such as rounded corners, gradients, and shadows;
- serve resized images, depending on screen size;
- reduce image size by adjusting compression and quality;
- combine or "sprite" images for serving through CSS; and
- use Gzip compression for code.

Use caching

HTML5's new abilities allow the persistent caching of files and data that survives browser sessions and power cycles. Designers can use these

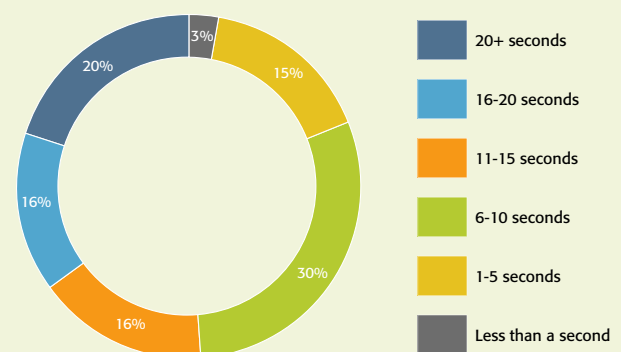


Figure A. Correlating page load time and user patience. Most users will wait only 6 to 10 seconds for a site to load. (Data from <http://blog.kissmetrics.com/loading-time>.)

features to reduce the latency time associated with downloading files on mobile networks, as well as to enable the site's offline capabilities.

To use caching techniques effectively, designers can

- use HTML5 `localStorage` to cache JavaScript and CSS files,
- use a naming convention to handle file versioning for cache refreshes, and
- implement server-side session and output caching as appropriate to minimize database calls and code-processing time.

Make speed a core design feature

When speed is a core feature in website creation, designers are more likely to make sound performance-optimization decisions. Two useful strategies are to

- consider a minimalist, image-light design that uses typography to increase readability and
- develop a dedicated site with minimum requirements for mobile use, allowing the removal of the code for advanced desktop interactions.

Test and refine

Many tools measure site performance and observe the impact of speed optimizations. Closely observing a breakdown of all page elements, including number of requests, file size, and response time will help designers focus on improving areas that will have the greatest impact on speed. Some popular tools include

- PageSpeed (<http://code.google.com/speed/page-speed/>),
- yslow (<http://developer.yahoo.com/yslow/>), and
- Blaze Mobile (<http://blaze.io/mobile>).

References

1. *How Loading Time Affects Your Bottom Line*, KISSmetrics Marketing Blog; <http://blog.kissmetrics.com/loading-time>.
2. G. Linden, *Marissa Mayer at Web 2.0*, Geeking with Greg, Nov. 9, 2006; <http://glinden.blogspot.com/2006/11/marissa-mayer-at-web-20.html>.
3. M. Cutts, *Google Incorporating Site Speed in Search Rankings*, Matt Cutts: Gadgets, Google, and SEO, Apr. 9, 2010; www.mattcutts.com/blog/site-speed.
4. J. Weinstein, *Let's Make the Web Faster*, Google Code; <http://code.google.com/speed/articles/mobile.html>.
5. B. Nadel, "3G vs 4G: Real-world Speed Tests," *Computerworld*, Dec. 16, 2010; www.pcadvisor.co.uk/news/internet/3253778/3g-vs-4g-real-world-speed-tests.
6. *Mobile Web Performance Optimization Tips and Tricks*, Blaze Software, Inc., July 2011; www.slideshare.net/blazeio/mobile-web-performance-optimization-tips-and-tricks.
7. Marc, *US Cellular 3G Speed: 2011 Mobile Broadband Review*, Apr. 24, 2011; www.mobile-broadband-reviews.com/us-cellular-3g-speed-2011.html.



Figure 4. Layout of the Noblis Innovation and Collaboration Center Demos website. Noblis designers used media queries to enhance touch-based navigation, placing buttons either on (a) the right side in landscape orientation or (b) on the bottom in portrait orientation. The layout in (a) accounts for the right-handedness of most people, while the layout in (b) maximizes content width to enhance readability.

An excellent example of responsive web design's power to adjust layout is the Hicksdesign website (<http://hicksdesign.co.uk>). Its creators used a combination of fluid architecture and media queries to scale and adjust the site's layout.

A sample application

In 2010, Noblis sponsored Noblis Innovation and Collaboration Center Demos, an internal independent research project to develop a website that showcases innovative project work. Analysts designed the interface for intuitive touch-based navigation and identified the iPad as the target device. They then used media queries to enhance site navigation according to orientation. Figure 4 shows the two orientations. Because most people are right-handed, designers positioned site navigation along the screen's right side in landscape mode to make reaching the navigation with a thumb more user-friendly. In portrait mode, they moved navigation to the screen's bottom to maximize the width available for content and enhance readability.

Implementation considerations

Responsive web design's flexible architecture and powerful media queries have many advantages, but they come with some implementation issues. A major one is that older browsers don't support media queries. However, nearly all newer versions offer full support (www.caniuse.com/css-mediaqueries).

Fortunately, browsers that do not support media queries will ignore them, which means that it's safe to introduce media queries regardless. To fill in browser support gaps, developers can use JavaScript libraries such as Respond (<https://github.com/scottjehl/Respond>) and Adapt.js (<http://adapt.960.gs/>), which mimic media query behavior by detecting browser width and loading conditional stylesheets.

Another implementation issue concerns scaling media-heavy websites for smartphones. In such cases, responsive web design techniques might not be ideal. Mobile users generally want to access information or perform a task as quickly as possible and

then move on. Because smartphones rely primarily on cellular networks with limited bandwidth for Internet connection, downloading a site's full version adjusted for mobile viewing can lengthen page load time.

Consequently, unless a website is built for speed from its conception, a dedicated mobile site serving optimized content is still the best way to achieve the speed that mobile users prefer. The sidebar "Optimizing for Speed" on p. 17 gives some guidelines and best practices for architecting a site for speed.

Next steps

Organizations that want to incorporate responsive web design into their current website design practices can take several immediate steps.

Examine existing websites

Many businesses are using this methodology to create dynamic user experiences, and examining these sites can help spur design ideas. About.com, which provides original content and advice on everything from home repair to the weather, recently introduced a new responsive design. The site features a fluid architecture with flexible images and adjusts to four possible layouts, depending on browser window size.

Users can see the responsive behavior by previewing the site in a modern browser such as Internet Explorer 9 or Google Chrome and then resizing the browser window. Other effective responsive web design implementations are the Charlotte Area Transit System (CATS) Clean Air Commute Challenge website (<http://clearairchallenge.com/>) and French World Wildlife Foundation's Earth Hour website (<http://earthhour.fr/>).

Target tablets

Introducing media queries that target tablets is one way to start experimenting with responsive web design. The iPad has a specific device width and supports media queries for orientation,

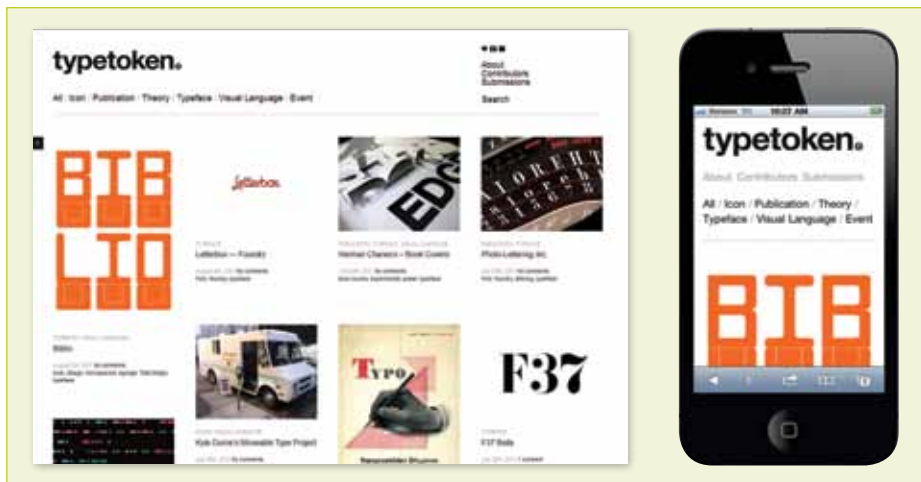


Figure 5. Implementing responsiveness in a minimalist design. Typetoken.net designers implement navigation elements with type, increasing the emphasis on the content images of the site. Using media queries and a fluid grid to adjust font-size and layout, the site easily adapts from a one column layout for mobile devices, to up to 10 columns for widescreen monitors up to 3000px in width. (Image courtesy of Typetoken; www.typetoken.net.)

so designers can add usability enhancements for its touch based interface without affecting the rendering on any other platform. Such enhancements might include increasing the size buttons and navigation links for easier touch selection or displaying content aimed at mobile users.

Make all design aspects responsive

Certain web design techniques and decisions increase the effectiveness of responsive web design. CSS3 capabilities enable visual elements such as rounded corners, gradients, and drop shadows. By implementing visual elements with code instead of images, designs can have the same look with a smaller page and fewer server requests, maximizing site speed.

CSS3 also makes it easier to design sites that scale visually. Scaling stylish content boxes with rounded corners used to require methods, such as the sliding door, which require extra markup for wrapping containers and images for each corner. With CSS3, this extra markup is no longer needed.

Minimalist design—reducing a design to its most important elements—is another concept associated with responsive design. Minimalist design often relies primarily on typography and spacing, not images, to enhance readability. CSS3's `@font-face` property provides a greater selection of fonts for web use, making organizational branding easier and reducing the design's need for flexible images. The screen shot of typetoken.net in Figure 5 is an example of how a minimalist design implements responsive design techniques. [Smashingmagazine.com](http://smashingmagazine.com) shows more minimalist design examples (www.smashingmagazine.com/2010/05/13/principles-of-minimalist-web-design-with-examples).

Consider mobile-first design

Considering a mobile-first design approach for planned sites and enhancements will increase the effectiveness of responsive web design techniques. Architecting for mobile platforms first, then using media queries to add style enhancements and high-resolution media will ensure a site optimized for speed on mobile

platforms that can scale up to devices with greater capabilities. Luke Wroblewski, an internationally known product designer, offers a wealth of information on his blog entry about mobile-first design and its numerous benefits for website development.⁸

Developing websites based on fixed assumptions such as minimum screen resolution ignores the variability inherent in the current digital landscape. With the ability to introduce styles that serve particular resolution and orientation changes, developers can now build sites that dynamically respond to viewing contexts across a spectrum of digital devices. Businesses can benefit from designing modifications to enhance the user experience for different devices instead of taking a “one size doesn’t quite fit all” approach. Although not a perfect solution to achieve the One Web vision, responsive web design is one way to reduce costs while enhancing the user experience. ■

References

1. *Scope of Mobile Web Best Practices*, W3C Working Group, Dec. 20, 2005; www.w3.org/TR/2005/NOTE-mobile-bp-scope-20051220.
2. *Mobile Web Best Practices 1.0*, W3C Working Group, July 29, 2008; www.w3.org/TR/mobile-bp/#OneWeb.
3. E. Marcotte, *Responsive Web Design*, A List Apart, May 25, 2010; [www.alistapart.com/articles/responsive-web-design](http://alistapart.com/articles/responsive-web-design).
4. E. Marcotte, *On Being “Responsive,”* Unstoppable Robot Ninja, Sept. 14, 2010; <http://unstoppablerobotninja.com/entry/on-being-responsive>.
5. E. Marcotte, *Fluid Images*, Unstoppable Robot Ninja, Apr. 17, 2009; <http://unstoppablerobotninja.com/entry/fluid-images>.
6. P.J. Lynch and S. Horton, *Web Style Guide-Basic Design Principles for Creating Websites*, 3rd ed., Yale Univ. Press, 2009.
7. S. Chuan, *Revisit Hardboiled CSS3 Media Queries*, Mar. 21, 2011; www.blog.highb.com/mobile-2/revisit-hardboiled-css3-media-queries.
8. L. Wroblewski, *Mobile First*, Nov. 3, 2009; www.lukew.com/ff/entry.asp?933.



Brett S. Gardner is a lead staff member at Noblis, where his experience includes software product management and development. He received a dual BS in management information systems and finance from Salisbury University. Contact him at brett.gardner@noblis.org.

The What, Why, and How of Mobile Applications

Daniel Y. Na

Evolving a mobile business strategy demands a multifaceted perspective. Not only must organizations identify effective processes, but they must understand and capitalize on the methods and technologies that will shape mobile development's future.

Just as the Internet sparked the rise of e-commerce in the 1990s, so the smartphone's popularity is revolutionizing business products and services worldwide. With online retail spending reaching a record of \$43.4 billion by 2010¹ and eBay reporting an increase in global mobile sales from \$600 million in 2009 to \$2 billion in 2010,² businesses are redoubling their efforts to gain access to and traction with a variety of consumers. For most organizations, the question is no longer if they should get involved but rather how quickly they can become part of the mobile market. Both developers and their target users have many choices in cellular service carriers, devices, and mobile operating systems (OSs), which means that a mobile business strategy must reflect multiple perspectives.

Starting with how to provide a rich and satisfying user experience, organizations must next identify the best application

development processes as well as capitalize on the methods and technologies that will drive mobile development into the future. Forward-thinking companies are already addressing the problems that will define the next market, such as focusing development on mobile web applications instead of on increasingly restrictive native applications. Looking at the mobile application market's current state, including its limitations, in light of development hurdles that these companies are overcoming is a first step toward identifying practical strategies to reach users and gain a foothold in the mobile universe.

Growth and obstacles

With 14.7 million devices by 2010 and a yearly growth rate of 41 percent, the U.S. smartphone market is the largest in the world.³ As Figure 1 shows, three mobile OSs hold roughly 66 percent of the overall market share—Apple iOS, Google Android, and Windows Phone 7 (WP7)⁴—and by 2015 the collective worldwide market share of these three OSs is projected to grow to more than 80 percent.⁵ Accordingly, the development and sale of mobile native applications (applications that users can download and install directly to their phones) is growing exponentially. It is not surprising then that mobile application downloads across all devices are expected to reach 50 billion by 2012, a sharp increase from the 7 billion such downloads in 2009.⁶ Figure 2 shows that growth from the perspective of Internet Protocol traffic.

Cross-platform complications

As “Bridging the Mobile App Gap” on p. 27 describes, creating applications for mobile devices presents a unique set

Inside Track

- Native mobile applications can suffer from disadvantages that mobile web applications avoid, such as increased development and maintenance costs, hardware incompatibilities, and platform-specific policy.
- Standards-compliant web browsers provide stable functionality for both the users and developers of websites and web applications, providing a consistent user experience regardless of operating system or device.
- Several major businesses have successfully deployed mobile web applications over native applications, including 37signals' BaseCamp and The Financial Times on the iPad.
- The consistency and accessibility of mobile web applications across devices increases customer reach and engagement, which in turn increases brand recognition for an organization.

of development challenges. Each of the three major mobile OSs require a different programming language for applications: Objective-C for iOS, Java for Android, and C# for WP7. An organization that aims to provide an application accessible across all three OSs must invest in the tools, resources, and expertise necessary for development in each of these disparate languages. Faced with theoretically triple the initial development costs for a single application, plus OS-specific maintenance and update costs, many organizations are boxed into a single platform, thereby losing customers that use other systems.

Further complicating cross-platform application development efforts are the differences in policy and hardware surrounding mobile devices from different manufacturers and cellular service carriers. For example, carriers offer a variety of phones running Android and WP7. Although such hardware variety is great for consumers, it can be a challenge for developers to accommodate unknowns such as screen size, input mode, storage capacity, RAM, and processing power. Furthermore, both carriers and manufacturers tend to push Android updates to users at different rates.⁷ Because OS changes can require changes to application code to maintain functionality, developers are forced to simultaneously accommodate legacy and the latest mobile technology.

Platform-specific policy

Selecting an application platform requires organizations to accommodate both the technical obstacles and business policies unique to that platform. Such accommodations are fine if initial obstacles and policies remain constant throughout an application's life cycle, but when platform owners unexpectedly enact changes that alter initial assumptions, developers are unprotected. Apple's unexpected alteration of its stated business model and pricing strategy is a good example.

In 2007, Apple introduced the iPhone and iOS, pioneering the successful use of touchscreen technology with an intuitive user interface. By 2011, over 70 million iPhones had been sold,⁸ and the seamless integration of the iPhone with the iTunes Music Store and Apple App Store has strengthened the iPhone's popularity, with over 15 billion songs and 14 billion applications downloaded as of June 2011.⁹ The commercial success of the iPhone as a mobile device has drawn both consumers and businesses alike to the iOS platform, each looking to capitalize on the development of iPhone-specific applications that suit their personal and business needs.

In February 2011, Apple elected to change its subscription cost model within the Apple App Store,¹⁰ which surprised and angered many mobile application developers. Apple implemented a 30

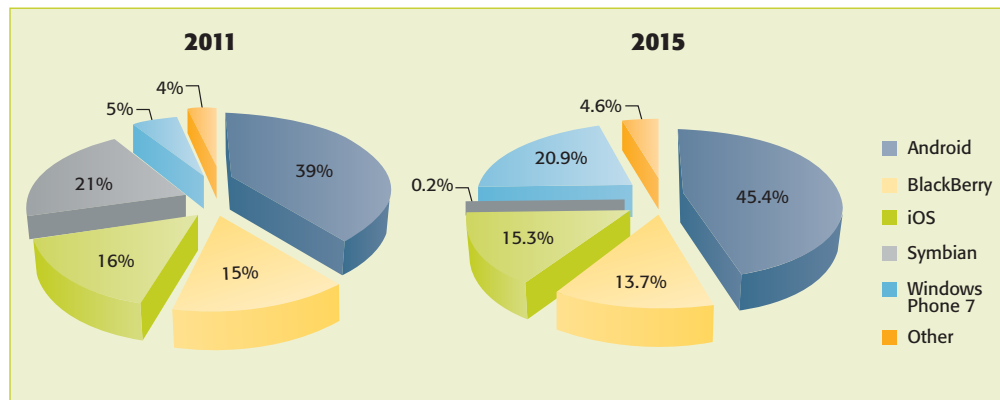


Figure 1. Mobile market share in 2011 and projections for 2015. According to IDC, Android, Windows Phone 7, and iOS are expected to continue to dominate the market. Symbian enjoyed its own growth until February 2011 when Nokia switched to Windows Phone 7. (2015 data from *IDC Forecasts Worldwide Smartphone Market to Grow by Nearly 50 Percent in 2011*, Mar. 2011; www.idc.com/getdoc.jsp?containerId=prUS22762811.)

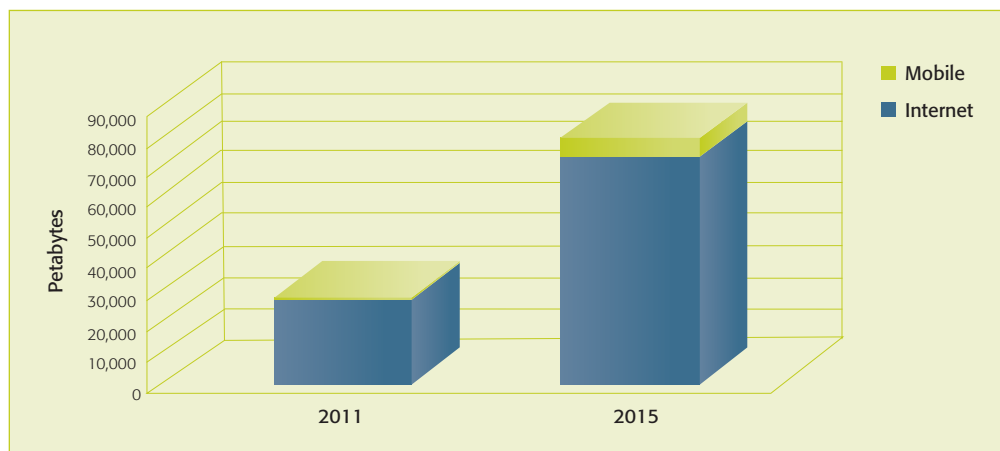


Figure 2. Current and projected global Internet Protocol traffic. Petabytes of mobile traffic are expected to grow from two to eight percent of global Internet Protocol traffic in the next four years. This growth translates to an actual increase in mobile traffic of more than 1,000 percent. (Data from *Cisco Visual Networking Index: Forecast and Methodology, 2010–2015*, Table 2, June 2011; www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html.)

The Language of Web Development

Here are some terms that mobile web application developers should be familiar with:

Ajax. Short for Asynchronous JavaScript and XML, Ajax is a group of interrelated web development methods used on the client side to create interactive web applications. With Ajax, web applications can send data to, and retrieve data from, a server asynchronously (in the background) without interfering with the existing page's display and behavior.

CSS3. The third and most recent iteration of the Cascading Style Sheets, CSS3 is a style-sheet language used to describe the presentation semantics (look and format) of a document in a markup language. CSS3's most common application is to style web pages written in HTML and XHTML. Like its CSS predecessor, CSS3's primary aim is to enable the separation of document content (written in HTML or a similar markup language) from document presentation, including elements such as the layout, colors, and fonts.

HTML5. As the latest version of the Hypertext Markup Language, HTML5 is used to structure and present web content. As of June 2011, HTML5 is still under development. Its core aims are to improve the language with support for the latest multimedia while keeping content easy to read for users and consistently understandable to devices (computers, web browsers, parsers, and so on).

jQuery. A lightweight cross-browser JavaScript library, jQuery emphasizes interaction between JavaScript and HTML. Used by 27-plus percent of the 10,000 most visited websites, jQuery is the most popular JavaScript library currently in use.

Web standards. This is the general term for the formal standards and other technical specifications that define and describe various web aspects. In recent years, the term has been more frequently associated with the trend of endorsing a set of standardized best practices for building websites and a philosophy of web design and development that includes those methods. The World Wide Web Consortium (W3C) determines web standards.

Webkit. The webkit is a layout engine that lets web browsers render web pages. It provides a set of classes to display web content in windows, and implements browser features such as following links that the user clicks on, managing a back-forward list, and managing a history of recently visited pages.

Source: <http://en.wikipedia.org>

percent fee on any iOS applications that required a subscription, and forced application developers to process all payments through Apple instead of managing payments on their own sites. Any software as a service or web application with a native iPhone application was forced to provide a 30 percent fee to Apple for both the initial sale on the App Store and any recurring payments in perpetuity. Companies could no longer simply redirect users to a signup form on the Web. Rather, developers had to provide that same signup mechanism in the application (in-app) for the same cost, despite a 30 percent loss of revenue. Apple planned to enact these changes on June 30, 2011.

Meanwhile, developers hotly debated the ethics of this new policy. As the owner of a highly successful mobile platform,

Apple has the legal right to extract revenue from that platform as it sees fit. However, many felt that third-party applications had a strong hand in fostering Apple's continued success. Several successful businesses with applications on the iOS platform noted that they could not sustain the unanticipated 30 percent decrease in revenue and remain financially solvent. Rhapsody, the number one subscription based music service in the United States, began exploring legal action.¹¹ In effect, Apple's changed policy was forcing organizations into a lose-lose position: abandon a market with millions of iPhone users or sacrifice a significant amount of revenue to pay the 30 percent fee.

Fortunately, Apple decided to revisit its planned revision.¹² In early June 2011, the company revised its plan to allow publishers to continue selling digital subscriptions outside their applications without providing a fee to Apple as long as there is no in-app button or link that redirects users to an external purchase page. The new policy also removes price restrictions, which means publishers can set in-app prices to accommodate the 30 percent premium that they must pay Apple.

The changes in Apple's App Store policy reflect a broader concern about product and business autonomy for application developers on all platforms. As long as mobile applications continue to be tied to specific distribution channels and OSs, organizations that want to reach those users will have to comply with specific policies and fees that they cannot control, or they must seek a suitable alternative.

Mobile web applications

With mobile web growth more than eight times that of desktop web growth and smartphone sales expected to pass PC sales in 2011,¹³ organizations are recognizing that maintaining positions of leadership requires prioritizing mobile technologies now. Many are already on a quest for the best way to overcome mobile's technical and policy obstacles without huge transition expenditures. Developing cross-platform mobile web applications with a single codebase is one strategy that allows organizations to stay ahead of the mobile curve.

Browsers and standards

Although differences persist in the implementation and technical specifications of iOS, Android and WP7, one commonality is the default inclusion of standards-compliant web browsers. Web standards are a formal set of recommendations that seek to "promote fairness, responsiveness, and progress" by defining the baseline of functionality that browsers are expected to support.¹⁴ The use of standards-compliant web browsers provides a foundational functionality for both the users and developers of websites and web applications, which in turn provides a consistent user experience regardless of OS or device.

In that sense, mobile browsers are the great equalizer in the realm of mobile technology. The browsers for iOS (Safari), Android,

and WP7 (Internet Explorer 9) all support web standards.¹⁵ Similarly, increasing mobile support for web technologies such as jQuery, CSS3 and HTML5 adds the necessary functional, aesthetic and multimedia tools for developers to mimic the behavior of native applications on the Web. The sidebar “The Language of Web Development” briefly describes these technologies and their primary roles in the mobile web.

By shifting the mobile paradigm to focus on the creation of robust, cross-platform web applications in place of native applications specific to each OS, businesses will save on development costs while reaching a wider customer base than ever before.

Open web standards

The prospect of unilateral policy determination by Apple, Google, or Microsoft looms unavoidably over organizations that build native applications for the platforms that each company owns. Fortunately, the beauty of web-based application development is that organizations simply need to pay attention to web standards development to be aware of what they can and can't control. Any device that accesses the Web serves its interests by conforming to web standards because these standards are the guidelines for building any website. Conformance also protects against a carrier's price or functionality discrimination. As long as the Web remains open, web application access will also remain open, and organizations will be free to establish their prices and functionality as they see fit.

Migration to web applications

The move from native to web applications is an extension of the trends in traditional desktop platforms. The proliferation of devices—desktops, laptops, tablets, smartphones, and so on—necessitates data access that is independent of device and platform. Consequently, data access and manipulation commonly occurs in the cloud (across servers rather than local machines), which renders local storage unnecessary and even unfavorable for many applications and further diminishes the need for native applications.

By publishing to the Web, organizations can avoid several obstacles that desktop or native software releases typically encounter:

- pushing updates is not contingent on user downloads and installations,
- organizations need not simultaneously support several versions of the same application, and
- organizations can provide support independent of the device's OS.

Recognizing these benefits, many major businesses are creating web applications that successfully fit their business needs. One of the first and most successful is 37signals' mobile interface to its BaseCamp web application. Since 2004, BaseCamp has been a leading web-based project management

and collaboration tool, with more than three million users worldwide. In early February 2011, 37signals decided to launch BaseCamp Mobile, a web-based and HTML5-driven version of BaseCamp.

The company's decision to create a mobile web application instead of a native application was motivated by several of the mobile development obstacles described earlier. In the 37signals blog, a company representative wrote,¹⁶

“Do we want to have to hire an iOS developer and an Android developer? That's a lot of specialization, and we're usually anti-specialization when it comes to development ... Plus, since WebKit-based browsers were making their way to the webOS and BlackBerry platforms too, our single web-app would eventually run on just about every popular smartphone platform.”

Another company, *The Financial Times* (FT) decided in June 2011 to launch an HTML5 web application instead of a native iOS application in part because they anticipated changes to the Apple App Store's terms and conditions. According to Rob Grimshaw, managing director of FT.com, the FT mobile web application allows FT to save the cost of building unique native applications for different mobile operating systems in addition to avoiding compliance with Apple's 30 percent requirement or any other terms and conditions. Figure 3 shows FT.com's invitation to iPad and iPhone users. Grimshaw noted, “There isn't a single feature in the native app we haven't been able to replicate in the Web app.”¹⁷

According to Forrester Research, the FT mobile application “lets [FT] distribute a digital version of its content to multiple devices and platforms while retaining control of the customer relationships.”¹⁸ Faced with what many publishers consider “punitive terms and conditions of some platform providers,”¹⁸ many industry-wide are eager to see if the FT model proves successful for FT and its over 3.6 million registered users.



Figure 3. FT.com's invitation to switch to a web application. By choosing web application development over the traditional native development path, FT saves the cost building unique applications for different mobile OSs. (Image courtesy of FT.com; *The New FT App for iPad and iPhone*, July 2011; <http://apps.ft.com/ftwebapp/>)

Limitations

Although mobile web applications have compelling strengths, they have several functional limitations that organizations must consider when deciding how to establish a mobile business strategy. Security is also an ongoing concern, as the sidebar “Growing Concerns over Mobile Security” describes.

Data-intensive calculations

Although the speed of mobile data access is accelerating, bandwidth limitations might require handling data-intensive calculations and processes within native applications. However, cellular network speed doubled from 625 kbps in 2009 to 1040 kbps in 2010 and is forecast to grow to 2.2 Mbps by 2015—a 10-fold increase.¹⁹

With a sufficiently fast data connection, data-intensive processing can be moved from client- to server-side devices. Consequently, the growth pattern of cellular data access not only supports the eventual elimination of bandwidth limitations but also strongly implies that native and mobile web applications will at some point have a very similar runtime.

Browser functionality

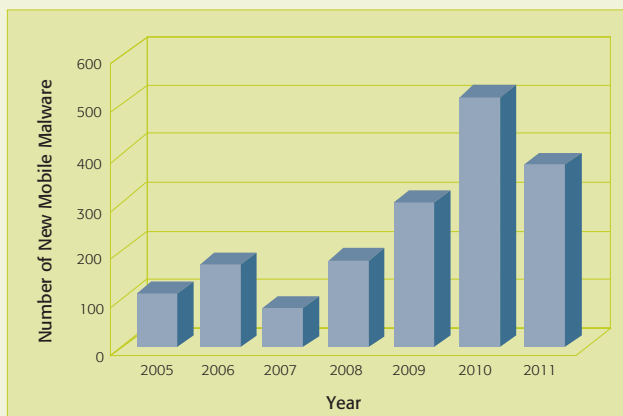
At present, mobile web applications are most hampered by the functional limitations of mobile browsers. Mobile devices provide unique capabilities that are atypical of desktop devices, such as Bluetooth, cameras, accelerometers, global positioning system, and telephony. Many native mobile applications exploit these functions, including photography and location services and games. However, web standards—the common denominator for mobile web applications—are the same across desktop and mobile devices. Because desktop users don’t often require accelerometer or location services, the browsers don’t commonly provide these functions. Consequently, mobile web applications are limited to the functionality of a desktop device’s browser, and developers must natively code applications that require access to phone-specific functions.

Several other browser-specific limitations within mobile OSs have yet to be resolved. For example, HTML5 audio controls in both the default Android browser and Mobile Safari are severely limited. Moreover, Mobile Safari doesn’t support file uploads within the browser, so web applications that require uploads must find suitable workarounds. Luckily these limitations are

Growing Concerns over Mobile Security

The popularity of smartphones and mobile applications has revolutionized how consumers discover, search, play, and shop on the go. But an unfortunate byproduct to the convenience and prevalence of mobile transactions is the increasing presence of mobile theft and software attacks on mobile devices. Figure A shows the growth in newly identified malware threats for the past seven years.

Much of the vulnerability stems from the unique characteristics of smartphones, such as always-on connectivity and cloud synchronization, which can exacerbate the ease by which threats spread.¹ According to Gustavo de los Reyes, executive director for AT&T Security R&D, “These phones are being used frequently for sensitive transactions like banking, mobile payments, and transmitting confidential business data, making them attractive targets if not protected.”²



Types of mobile threats

Mobile threats vary in scope and severity, including malicious applications, botnets, spyware, and phishing.¹ *Malicious applications*, which users inadvertently download to their phones, contain code to steal personal information, including logins and passwords for a variety of services. Such applications can also install additional malicious programs, such as botnets, without the user’s knowledge.

Botnets, multiple infected machines working together to perform harmful acts, are commonly introduced on mobile devices via email attachments, malicious applications, infected websites, or some combination. Botnets derive their power from working with other infected phones, and consequently actively seek to propagate themselves to contacts and synced devices.

Spyware is another common mobile threat familiar from its pervasiveness in the desktop world. Spyware, which is available online, monitors, records, and transfers communications data from a mobile device to an external source. Communications data includes phone calls, emails, text and media messages, GPS location data, photos, and more.

Phishing, or attempting to extract sensitive personal data from users through fake login screens or notification messages, poses a significant problem on mobile devices. Users tend to trust their mobile devices more than their computers, yet mobile phishing now occurs not just through email or websites but also through avenues specific to mobile devices, as text and multimedia messaging.

Figure A. Number of new mobile malware signatures added to antivirus databases from 2005 to 2011. The trend in mobile malware is increasing annually, as the 2011 figure reflects only the first half of the year. (Figure from Y. Namestnikov, *IT Threat Evolution for Q1-2011*, May 18, 2011; www.securelist.com/en/analysis/204792176/IT_Threat_Evolution_for_Q1_2011.)

neither insurmountable nor permanent. Experts are confident in a short timetable for significant HTML5 advancements. Indeed, AT&T executive Ted Woodbury predicts that in 18 months the shift to HTML5 will boost web application stores and “take a serious bite out of the OS-driven app stores.”²⁰

Going forward

The first developers to embrace the mobile platform profited by early adoption, capitalizing on their place as market leaders while mobile popularity skyrocketed. Their positioning provided value not only in technical experience, but also in business savvy when it came to selling products and services through an entirely new medium. Given an increasingly digital and interconnected world, it is essential to become familiar with mobile application development and to adopt strategies that avoid the technical and policy pitfalls of native applications, which can kill the viability of any mobile product.

In an era of multiple platforms, each with a strong following, many successful organizations are focusing on the development of standards-compliant web applications. Offering application

access through the mobile web affords protection from policy or purchase restrictions and enables the use of a single code base for all devices. Not only does such an approach lower hiring, development, deployment, and support costs, but it also frees the organization to concentrate resources on web-related technologies. “Giving Data a New Face” on p. 33 offers some specific development techniques and strategies.

Mobile web applications also provide platform independence, which gives organizations a greater chance to reach the widest possible customer base, thereby maximizing revenue opportunities. A less obvious but equally important benefit of wider distribution channels is developing a positive user experience, which reinforces brand loyalty. Blogs that praise and condemn applications by turn have become pervasive, often reporting the positive and negative details of application use. Users are quick to appreciate assurances of the continued access and reliable operation of their favorite apps, which can boost customer satisfaction and brand recognition. Social media is a powerful marketing tool in this context. By leveraging mobile-specific capabilities such as location-based services, organizations can integrate applications with social media

User self-protection

Several traditional computing security measures are available on mobile platforms. Many of these products are not yet fully matured or cost effective for individual users, but the demand for protection is expected to grow, which should fuel the refinement of safety products accordingly. Table A shows two product categories integral to mobile security and their vendors.²

Operating system security

Developers of all the major mobile operating systems have built in four security features: process and file system isolation; app or code signing; ROM, firmware and factory restore; and kill switches. Process and file system isolation means that applications are limited in what phone functions they can access, while app or code signing means that some platform owners (such as Apple) can choose to explicitly approve or deny applications before they are able to be installed on user devices. Figure B shows how much control each major platform exercises in approving applications.

ROM, firmware and a combination of kill switch and factory restore give users ultimate control over what applications are running on their device. Using a kill switch, a user can force-quit a malicious or frozen application and then engage a factory restore to reset the infected phone to its original state.¹

Table A. Safety products available for mobile devices.

Product Category	Representative Vendors
Antivirus and firewall	Fortinet, F-Secure, Juniper Networks, Kaspersky Lab
Mobile encryption	SecurStar, Credant Technologies

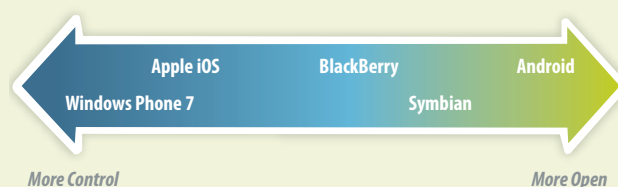


Figure B. Continuum of control exercised by platforms in approving applications for distribution. Windows Phone 7 and Apple iOS are the most restrictive of what applications are available to users, while Android is the most open.

Businesses, cellular service carriers, and hardware manufacturers are also taking action to protect users on a larger scale. For example, businesses can leverage the capabilities of security vendor MobileIron to deliver applications directly to employees, giving the businesses themselves direct control over the security of installed programs. Carriers such as AT&T and Verizon are introducing mobile security suites for their users in 2011, and several hardware manufacturers (Freescale Semiconductor, IBM, Intel, LG, Motorola, and Nokia) are working together to produce hardware with built in encryption mechanisms.¹

References

1. D. Barrera and P. Oorschot, “Secure Software Installation on Smartphones,” *IEEE Security & Privacy*, vol. 9, no. 3, pp. 42–48, June 2011.
2. N. Leavitt, “Mobile Security: Finally a Serious Problem?” *Computer*, vol. 44, no. 6, pp. 11–14, June 2011.

to communicate with customers within specific locales and increase customer interaction and engagement.

Although current versions of mobile browsers have some functional limitations, the rapid development of mobile technologies will soon overcome any such obstacles. Few would debate that mobile is the largest and fastest growing segment of the communications market. If businesses adopt the mobile web now, all stakeholders in the mobile application lifecycle will reap the benefits. Users will receive wider access to services, businesses will generate more revenue while limiting costs, and developers will be able to provide a consistent user experience across devices from the same code repository. Much like the Internet age of the 1990s, the mobile age—with all its benefits and challenges—has arrived. ■

References

1. L. Rao, *Online Retail Spending Reaches a Record \$43.4B in Q4 2010; Up 11 Percent*, Feb. 2011; <http://techcrunch.com/2011/02/04/online-retail-spending-reaches-a-record-43-4b-in-q4-2010-up-11-percent/>.
2. R. Wauters, *eBay: Mobile Sales Grew from \$600 Million to \$2 Billion in 2010*, Jan. 2011; <http://techcrunch.com/2011/01/06/ebay-mobile-sales-2010/>.
3. Canalys, *Android Smartphone Shipments Grow 886 Percent Year-on-Year in Q2 2010*, Aug. 2010; www.canalys.com/pr/2010/r2010081.html.
4. C. Ziegler, *Visualized: US Smartphone Market Share, by Manufacturer and Platform, Made Pretty*, Mar. 2011; www.engadget.com/2011/03/03/visualized-us-smartphone-market-share-by-manufacturer-and-plat/.
5. IDC, *IDC Forecasts Worldwide Smartphone Market to Grow by Nearly 50 Percent in 2011*, Mar. 2011; www.idc.com/getdoc.jsp?containerId=prUS22762811.
6. S. Schroeder, *Mobile App Market to Surge to \$17.5 Billion by 2012*, Mar. 2010; <http://mashable.com/2010/03/17/mobile-app-market-17-5-billion/>.
7. J.R. Raphael, *Android Upgrades: Which Manufacturers Can You Trust?*, Jan. 2011; http://blogs.computerworld.com/17649/android_upgrades.
8. Crunchdot, *Apple Sold 14.1 Million iPhones Last Quarter, Over 70 Million Since Launch*, Oct. 2010; www.crunchdot.com/apple-sold-14-1-million-iphones-last-quarter-over-70-million-since-launch/.
9. L. Rao, *Apple: 15 Billion Songs Sold, 130 Million Books, 14 Billion Apps Downloaded*, June 2011; <http://techcrunch.com/2011/06/06/apple-15-billion-songs-sold-130-million-books-14-billion-apps-downloaded/>.
10. Apple, *Apple Launches Subscriptions on the App Store*, Feb. 2011; www.apple.com/pr/library/2011/02/15appstore.html.
11. M. Rosoff, *Rhapsody Slams New Apple Subscription Rules*, Feb. 2011; www.businessinsider.com/rhapsody-slams-new-apple-subscription-rules-2011-2.
12. J. Golson, *Apple Reverses Course on In-App Subscriptions [Apple Confirms]*, June 2011; www.macrumors.com/2011/06/09/apple-reverses-course-on-in-app-subscriptions/.
13. ITU, M. Lipacis, and Morgan Stanley Research, *The Mobile Internet Report (Setup)*, Dec. 2009; www.morganstanley.com/institutional/techresearch/pdfs/2SETUP_12142009_RL.pdf.
14. World Wide Web Consortium, *Standards FAQ*; www.w3.org/standards/faq.
15. T. Ferrill, *Why You Should Care about IE9 on WP7*, Apr. 2011; www.wpcentral.com/why-you-should-care-about-ie9-wp7.
16. T. Mobily, *Launch: Basecamp Mobile*, Feb. 2011; <http://37signals.com/svn/posts/2761-launch-basecamp-mobile>.
17. S. Yin, *Financial Times Snubs Apple App Store with Browser-Based App*, June 2011; www.pcmag.com/article2/0,2817,2386596,00.asp.
18. N. Thomas, *The Financial Times Makes a Bold Move in Paid Apps*, Forrester Research, Inc., June 2011; www.forrester.com/rb/Research/financial_times_makes_bold_move_in_paid/q/id/60082/t/2.
19. Cisco, *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2010–2015*, Feb. 2011; www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html.
20. N. Gohring, *"HTML5 Not Yet Solving Mobile Dev Issues," InfoWorld*, May 2011; www.infoworld.com/d/mobile-technology/html5-not-yet-solving-mobile-dev-issues-268.



Daniel Y. Na is a staff member at Noblis, where his experience includes web application development and systems analysis and design. He received a BS in systems engineering and management information systems from the University of Virginia. Contact him at daniel.na@noblis.org.

Sigma—In Addition

More from Noblis authors:

Data Fusion and Collaboration: Noblis Responds to the IEEE VAST Challenge, C. Campbell, presented at Technology Tuesdays, Oct. 5, 2010; www.noblis.org/NewsPublications/Pages/CatherineCampbell.aspx.

"Open Source for Enterprise Geographic Information Systems," D. Garbin, J. Fisher, published in *IT Professional*, vol. 12, no. 6, pp. 38–45, Nov./Dec. 2010.

Visualization with Pivot Collections, M. Corrigan, presented at Technology Tuesdays, Nov. 2, 2010; www.noblis.org/NewsPublications/Pages/MileCorrigan.aspx.

"Latent Fingerprint Quality: A Survey of Examiners," R.A. Hicklin, et al., published in the *Journal of Forensic Identification*, 61(4), July 2011.

Method and System for Personal Information Extraction and Modeling with Fully Generalized Extraction Contexts, V. Pollara, US Patent 7,949,629, Feb. 15, 2011.

System and Method for Providing Network Resiliency, D. Garbin, US Patent 7,986,617, July 26, 2011.

"IEEE Open Source Software in Law Enforcement," M. Hon, G. Russell, M. Welch, published in *IT Professional*, vol. 12, no. 6, pp. 18–23, Nov./Dec. 2010.

Integrated Corridor Management: Analysis, Modeling and Simulation, invited speaker at the American Public Transit Association TransITech Conference, Session 9—Integrated Corridor Management, Miami, FL, Mar. 30, 2011; [http://apta.com/mc/transitech/schedule/Presentations/session%209%20Wunderlich%20Integrated%20Corridor%20Management%20Analysis%20Modeling%20and%20Simulation%20\(AMS\).pdf](http://apta.com/mc/transitech/schedule/Presentations/session%209%20Wunderlich%20Integrated%20Corridor%20Management%20Analysis%20Modeling%20and%20Simulation%20(AMS).pdf).





Bridging the Mobile App Gap

Adam M. Christ

Mobile application development used to mean writing code for each device platform, but emerging web technologies are spawning hybrid development approaches, which promise to change that fragmented picture

With the recent boom of smartphone and tablet market growth, mobile application development is quickly becoming an area that can no longer be ignored. According to the International Data Corporation, smartphones outsold PCs for the first time ever in the fourth quarter of 2010.¹ BusinessWire.com also reports that the worldwide smartphone market grew 79.7 percent from the first quarter of 2010 to the first quarter of 2011,² as older smartphone models became more affordable and vendors released highly anticipated models, such as the Verizon iPhone. Experts predict that this expansion rate will continue over the next few years, ensuring that the demand for mobile applications will escalate.³ For application developers, the question becomes how to reach this ever-expanding mobile audience.

Figure 1 gives a flavor of why mobile applications development can be intimidating. The staggering number of available devices has split the market among several mobile platforms, with the majority share belonging to Symbian, Android, Apple iOS, RIM

(BlackBerry) and Windows Phone 7. Because each platform is different, developing a native application—which requires a specific programming language and software development kit—typically means having to learn a development language and maintain a code base for each platform.

Apple's iOS, for example, is the platform for iPhone, iPad, and iPod. Developing an application for those devices requires the use of Xcode, a specific integrated development environment (IDE) and programming the application in Objective-C. To run the same application on Android, another popular platform, its developers must switch to the Java programming language and use the Eclipse development environment.

Having to learn multiple programming languages, software development kits, and IDEs, as well as managing several code sets creates considerably extra work, which translates to increased development time and a greater probability of code errors. These consequences, in turn, greatly increase the cost of the application's initial development as well as its post-release maintenance.

Fortunately, new technologies are changing this picture. Although native application development now dominates mobile strategy, recent innovations in web-based technologies are offering alternative strategies for developing applications across platforms using a universal code base. But perhaps the most important evolution is the hybrid approach, which seeks to close the gap between native and web-based application development, offering platforms that work with multiple operating systems and next-generation mobile devices. Given alternative web-based and hybrid application development approaches, breaking into the mobile application universe is no longer as hard as it once was.

Inside Track

- Developing native mobile applications requires the knowledge of multiple programming languages and separate code sets, resulting in high development time and cost.
- Advances in web technologies and device capabilities have created alternative application development technologies that are web-based, use proprietary middleware and clients, or are a hybrid of webkit and native applications. The hybrid approach provides the best trade-off.
- Platform selection and application compilation are only half the battle. A hybrid application must also deliver a user experience that complements a range of mobile devices and their features considering screen size and performance.
- Using an application platform and a mobile JavaScript framework, developers can create a hybrid application relatively quickly. The General Services Administration's Sustainable Facilities Mobile application is a case in point.

Development alternatives

Alternative application development technologies are web-based, use proprietary middleware and clients, or are a hybrid of webkit and native applications. A project's requirements will

PLATFORMS	 Apple iOS	 Android	 RIM (BlackBerry)	 Symbian	 Windows Phone 7	 Palm (webOS)
PROGRAMMING LANGUAGES	OBJECTIVE-C	JAVA	JAVA	SYMBIAN C++	C#	JavaScript, HTML & CSS
DEVICES (Only a small sample)						

Figure 1. Sampling of the mobile platforms and their various programming languages and devices. Multiple programming languages and diverse device specifications increase time and cost of development.

determine which alternative is the most suitable. However, of these alternatives, the hybrid approach, with its benefits of both web-based and native applications, generally offers the best trade-off of native application pluses and flexible retail options.

Web-based

Web-based applications use Hypertext Markup Language, Version 5 (HTML5), Cascading Style Sheets (CSS), and JavaScript to create mobile websites that look and feel like a native mobile application. Products from the accelerated smartphone market and the webkit-enabled browsers they contain are making such applications possible.

A typical web-based application uses a JavaScript framework like jQTouch or Sencha to replicate a mobile user interface. These frameworks, which are solely for mobile development, accommodate touch-enabled devices and use a website to deliver a mobile user experience complete with animations, offline capabilities, and even limited access to device features in some cases.

As such, web-based applications are essentially multiplatform, since viewing requires only a browser. The advantage of this alternative is its obvious convenience. The disadvantage is its product limitations; being web-based, it is not a candidate for other retail venues, such as an app store, possibly affecting availability and certainly limiting revenue production.

Proprietary middleware and clients

In this alternative, application development is based on prebuilt services like Red Foundry, which lets developers create mobile applications through a web interface by selecting prebuilt modules à la carte. There is thus no need to write code.

Module categories include but are not limited to audio, video, RSS, and photo. Each category then contains modules with different layouts and functionality based on that category. A user selects from the available modules and then, through web-based forms, customizes the module with video feed information, images, and so on. Once the application is complete, the service builds a native application that the developer can submit to an application store.

The advantage of this approach is the ease of creating native applications that look good and perform well. The down side is that functionality and design are limited to only what's offered, and generally prebuilt services come with a fee.

Hybrid webkit and native

Applications developed with a hybrid solution are native applications that use web technologies in place of a programming language like Objective-C. Hybrid applications are web-based applications built into native applications, so they possess the benefits of both. Most hybrid applications use platforms such as PhoneGap or Titanium to wrap HTML5, CSS, and JavaScript code to make several device features available through a JavaScript application programming interface (API), such as the camera, accelerometer, and storage.

Unlike the other alternatives which are confined to browsers or have limited functionality or preset features, the hybrid approach gives developers greater control over application design, yet allows access to device features. These benefits make it possible to develop a multiplatform application from a single code base and still have performance and availability comparable to that of a native application. Figure 2 shows how developers might use a hybrid solution in this way.

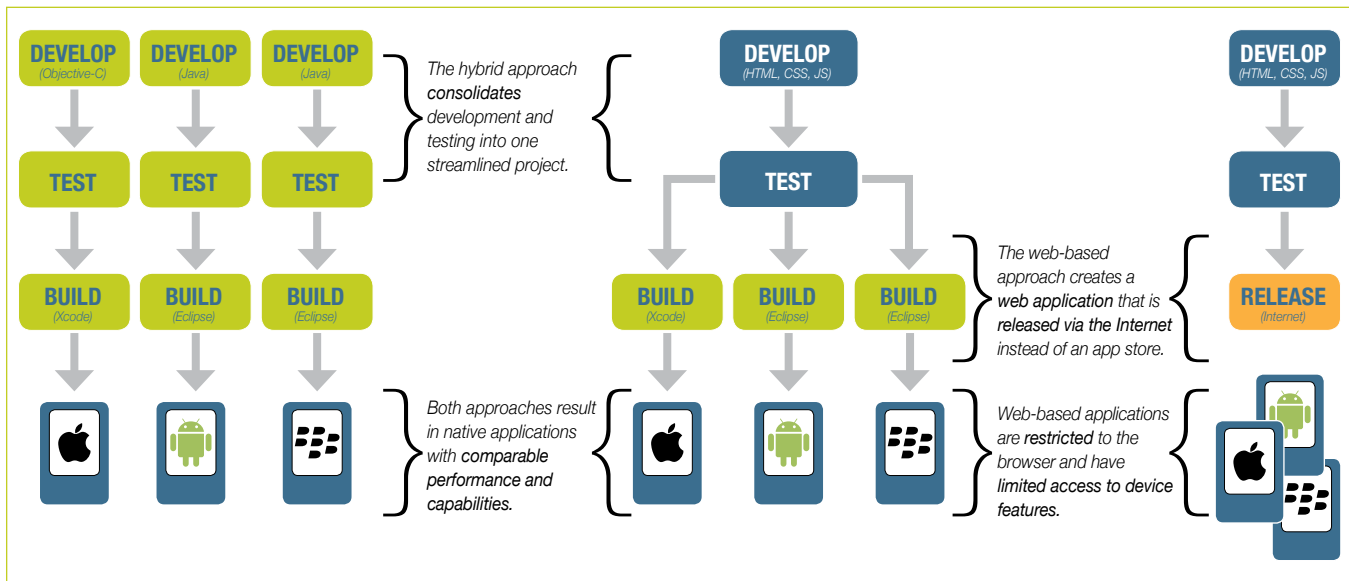


Figure 2. The continuum of mobile application development. Hybrid applications are closing the gap between native and web-based application development.

Elements of a hybrid solution

In implementing a hybrid solution, developers need to form a mobile strategy, carefully select the user-preferred platforms, and ensure that the choices reflect user desires and details that contribute to the total user experience. The sidebar “Before You Begin” outlines some basic mobile strategy steps. “Toward a User-Centric Digital Ecosystem” on p. 4 of this issue describes the characteristics of the total user experience in greater detail.

Application platforms

Table 1 gives a sampling of platforms and their supported mobile operating systems, how to acquire the necessary files or software, setup process, and cloud-based compilation availability. Any of these platforms will let developers create a mobile application using web technologies, and no single platform is best for all projects. Differences between each platform can make selection a daunting task, but further analysis of the features, requirements, and availability of each platform can bring to light which platform best suits a project.

PhoneGap. Nitobi developed PhoneGap as an open-source platform for creating native mobile applications using web technologies. Developers can download PhoneGap as a zip file that includes all the files needed to develop the application on each platform. A minor drawback is that each mobile platform requires a slightly different setup. iOS (Xcode) and Android are good examples. PhoneGap will integrate directly into Xcode, allowing developers to create a PhoneGap project from the new project dialog. For Android, however, developers must include the PhoneGap files in an existing project and update some of the project files. To address these setup differences, the company provides detailed tutorials for each platform on its website.

Once developers build the application using HTML5, CSS and JavaScript, they compile it into a native application using a build script or an IDE tailored to the mobile operating system. For applications targeting multiple mobile operating systems,

Before You Begin

Before starting any mobile application development, it is essential to outline a mobile strategy to help define the application’s scope, usability, and schedule. Although following these basic steps won’t guarantee a successful application, it can help you deliver an application on time that meets its target users’ needs.

1. Determine platform preference

Ask what platforms the application should support. Depending on the target devices, the application can be either cross platform or dedicated. Regardless of platform choice, a mobile website will ensure that all potential users can access content.

2. Develop use cases



Ask what user types are likely to need or enjoy the application and from there, determine what information or functionality the application must provide. Use cases can significantly aid in designing the user interface and determining the application flow so that users can quickly access what they need.

3. Create a release timeline

This strategy step is critical for a cross-platform application because releasing an application can be just as involved as developing one. A timeline helps prevent chaos and ensures that each application release for each platform is on schedule. It also helps in readying all documentation, images, and files in time for each release.

Most application stores or marketplaces require developers to submit the application for review, which can mean extra preparations. Because of this, releasing the application for one platform at a time can make the whole process smoother.

Table 1. Advantages and disadvantages of popular application platforms.

Platform	 PhoneGap	 Titanium	 appMobi{!}	 rhomobile
Advantages	Supports all popular mobile OSs, open source, cloud-based compilation (in beta).	Supports Apple iOS, Android and RIM (in beta); free basic membership and software.	Supports Apple iOS and Android; free XDK and initial application build; cloud-based compilation.	Supports Apple iOS, Android, RIM and Windows; open source; uses a model-view-controller framework; cloud-based compilation.
Disadvantages	Must install/use separate IDEs and software development kits; setup differs based on OS.	Limited mobile OS support; must install software development kits; no cloud-based compilation; additional support and training requires premium membership.	Limited mobile OS support; cost for post-release application updates; requires Google Chrome and Java 6 platform.	Requires installation and knowledge of Ruby; additional Rhomobile products require monthly membership.
Best Suited For	Projects that require greater control over setup/development and support a high number of mobile OSs.	Projects requiring quick setup and compilation.	Projects requiring a large number of testing users.	Projects looking for a structured framework.

both the setup differences and need to use several IDEs add work, but the extra effort is required only at the project's start and thus minimally impacts overall development time.

To further reduce development time, PhoneGap supports a build service that obviates the need for multiple software development kits and compilers. With this service, developers can upload an application consisting of HTML, CSS and JavaScript directly to the PhoneGap site, and cloud-based services compile the application. App-store-ready files can then be downloaded for release. As of June 2011, this service was in the beta stage, supporting only select mobile operating systems, and the company had not revealed if the service will require a fee.

Titanium. Unlike PhoneGap, Appcelerator's Titanium Developer is downloadable as a desktop program to build and compile applications. Titanium's strong advantage is that it eliminates the need to set up multiple IDEs. Developers code a web-based application using external editors, and Titanium performs like a slimmed down IDE, not only compiling the application but allowing its configuration and testing as well.

To build an application, developers first download and install platform-specific software development kits and then create an account. As of June 2011, Titanium supports only the iOS and Android platforms with BlackBerry support in beta. The basic development membership is free, and premium subscription-based memberships are also available, which provide training and certification, analytics, and other benefits.

AppMobi. AppMobi uses the appMobi XDK, a Java application that runs inside a Google Chrome instance, accessing a developer's local file system to create the necessary project directories and files. Developers configure and compile applications as they do with Titanium, but appMobi XDK provides a watered-down emulator that provides greater visualization of the applications.

AppMobi XDK also performs all compilation through cloud-based services and features local as well as over-the-air distribution to multiple devices, thus simplifying application testing.

As of June 2011, the company had no plans to support operating systems other than iOS and Android. Members can create unlimited applications for free; post-release updates and additional services are offered for a fee.

Rhomobile. Rhomobile provides a suite of tools for mobile application development with Rhodes at the center. Rhodes is an open-source framework based on Ruby. In contrast to other platforms, which require developers to write code into the application view as JavaScript, Rhodes supports a model view controller design, making it the most distinct of the platforms described. The Rhomobile suite also includes RhoSync for syncing data between applications and backend applications, RhoHub for cloud-based compilation, and RhoGallery for managing Rhomobile applications. Rhomobile products are

free to members. The only downside is the need to learn the basics of Ruby.

Creating a mobile user experience

Platform selection and application compilation is only half the battle. A hybrid application must also deliver a user experience that complements the mobile device's features. Part of ensuring a satisfying mobile experience is to understand how web and mobile applications differ. Consider viewing a website through a PC or laptop browser. Typically the website layout complements the display, making site navigation easy and intuitive. Images and fonts are the appropriate size for their function, and layout can be as complex or as simple as desired. But the same website viewed on a screen with a fraction of the former display's size and resolution will probably be nearly illegible. The mobile user must zoom to see fonts that were easily read on the larger display. Complex layouts with many pages require tedious minutes of scrolling. Navigation ranges from difficult to impractical. All of this creates a less-than-satisfying user experience.

Simply put, web layouts and designs do not translate well to mobile devices, which is why the user experience must be a crucial concern in any hybrid application development. The user interface and application flow must work with the device's screen size and perform well on a range of device sophistication. As smartphones and tablets become faster, device performance might become less of an issue, but until then developers must be aware of how the rules change from PC to mobile.

Over the past three years, open-source mobile JavaScript frameworks have made it possible to replicate the mobile experience using web technologies, taking care of most changes needed to use HTML controls on a mobile device. Typically these frameworks transform web-based controls and layouts by restyling them to be more suitable for mobile displays. Because their creators had touch displays in mind, they restyled and sized HTML controls and inputs specifically for ease of use on touch-enabled devices.

Among the many mobile JavaScript framework choices, a few of the most popular are jQTouch, jQuery Mobile, and Sencha Touch. The sidebar "Three JavaScript Frameworks" briefly describes the main characteristics of each. All mobile JavaScript frameworks are open source and have other strong advantages:

- **Development across platforms and devices.** Because each framework is designed to work across all popular platforms and devices, developers need only a single code base.
- **Touch-optimized layouts and user interface elements.** All layouts, form controls and widgets automatically adapt to varying device form factors.
- **Rich animations.** Smooth transitions between screens create a native application feel.
- **Themes.** Prebuilt themes use new CSS3 properties to provide strong visuals from the start.

Three JavaScript Frameworks

Three JavaScript frameworks are capturing the attention of the mobile development community, and any one of these might be suitable for a particular mobile application. However, each framework has its own set of quirks and blemishes, and as with web development, browser performance varies for the same code. Thus, it is crucial to test any hybrid application on every intended platform to ensure that all functionality is consistent and supported.

jQTouch

jQTouch is one of the pioneer mobile JavaScript frameworks. Built as a plugin for the popular jQuery JavaScript library, jQTouch provides basic user interface controls for a mobile application and uses iPhone-like animations to transition among screens. Because this framework was one of the first mobile JavaScript platforms, its features are limited relative to the other frameworks, but it is very polished.

jQuery Mobile

jQuery Mobile is another plugin built on the jQuery JavaScript library, but jQuery Mobile, being relatively nascent, supports all popular smartphone and tablet devices, providing the same user experience regardless of device. As of June 2011, the framework was in the early beta development stage. Like jQTouch, it offers restyled user interface controls and animations, and boasts a small file size.

Sencha Touch

Unlike jQTouch and jQuery Mobile, Sencha Touch uses the Ext JavaScript library instead of jQuery. It is object oriented and has more of a learning curve than the other two mobile frameworks. With Sencha Touch, display elements are more commonly created via the JavaScript code instead of HTML, which might appeal to more advanced developers.

- **Construction with application platforms in mind.** Some frameworks are designed to work hand in hand with platforms like Phonegap.

A hybrid development example

Using an application platform and a mobile JavaScript framework, developers can create a hybrid application relatively quickly. The General Services Administration's Sustainable Facilities Mobile (SF Mobile) application is a case in point. In early 2011, Noblis began development on a hybrid application as part of its independent research and development program in user experience design. The goal was to use alternative technologies to develop a mobile component for the General Services Administration's Sustainable Facilities Tool (SF Tool).

After researching multiple application platforms and JavaScript frameworks, the development team chose PhoneGap and jQTouch on the basis of the expected performance and ease of use with those platforms. With these technologies along with platform-specific IDEs and a custom API to provide the data, the team was able to fully develop SF Mobile in four months. The iOS application is free through Apple's App Store and Android Marketplace, and the plan is to release it for the BlackBerry (version 6) by the end of third-quarter 2011.

The success of this application is a good indication of the possibilities that hybrid applications can deliver for those already versed in web technologies. Free from the need to learn multiple software development kits, SF Mobile developers were able to capitalize on their existing knowledge to greatly reduce application development time. Informal testing on multiple devices confirmed performance differences between mobile systems, but also showed that the perceived differences minimally impacted the user experience. Once again, the choice of application platform and JavaScript framework was based on the application's nature. SF Mobile provides information, but had it been a game with a high graphical output or background calculations, another development alternative might have been more suitable. As application platforms, web technologies, and devices improve, the gap between native and nonnative applications will shrink allowing hybrid and other applications to perform more like their native counterparts.

Developing a mobile application used to be a huge hurdle requiring large amounts of preparation and development time, but new technologies and processes have drastically lowered that bar. Whether or not developers use these new alternatives or rely on native applications alone depends on the target users and project scope. Also alternative approaches are still in their infancy, so developers must be aware of the drawbacks inherent in any new process, such as inconsistent use of the approach; variations in performance, accessibility, and ease of use; and hardware differences.

Time will likely eliminate any obstacles. Many application platforms already implement or are working to implement cloud-based compilation to build applications for multiple platforms. As their creators enhance and refine these services, there will be less need to install multiple software development kits and IDEs, removing a chunk of the required platform setup time. Companies are already using these alternatives, primarily the hybrid approach, which even in its infancy, is a strong solution. When developers use the hybrid approach thoughtfully, the result can be comparable to a native application—a true bridging of the gap from native to web-based applications.

Application platforms, mobile JavaScript frameworks, and the devices themselves will become increasingly sophisticated, boosting both hybrid application performance and ease of development. For those who once feared the complexities and time-consuming process of building and releasing mobile applications, this is good news indeed. ■

References

1. S. Perez, "Smartphones Outsell PCs," *New York Times*, Feb. 8, 2011; www.nytimes.com/external/readwriteweb/2011/02/08/08readwriteweb-smartphones-outsell-pcs-74275.html?partner=rss&emc=rss.
2. "Smartphone Market Grows 79.7% Year Over Year in First Quarter of 2011, According to IDC," *Business Wire*, May 5, 2011; www.businesswire.com/news/home/20110505007011/en/Smartphone-Market-Grows-79.7-Year-Year-Quarter.
3. "IDC Raises Smartphone Growth Forecast," *Telecom Paper*, Sept. 8, 2010; www.telecompaper.com/news/idc-raises-smartphone-growth-forecast.

The trademarks and trade names appearing in this article are the property of their respective owners. The views expressed [in this article] are those of the author and not necessarily those of Noblis.



Adam M. Christ is a staff member at Noblis, where his experience includes web application development. He received a BS in science from Purdue University. Contact him at adam.christ@noblis.org.

Noblis Named to Ethisphere's List of
**World's Most
Ethical Companies**
for the
Third Time

2011 | **WORLD'S MOST
ETHICAL
COMPANIES**
WWW.ETHISPHERE.COM

This honor, awarded by Ethisphere Institute, honors companies that demonstrate **real and sustained ethical leadership** in their industries.



Giving Data a New Face

Seth A. Blanchard

Excel files and database records will remain the data infrastructure's backbone, but they do a poor job of transforming data into information. Data visualization makes it easier to see patterns, and Internet delivery can promote exchanges that lead to innovation.

Neither organizations nor individuals are coping gracefully with data glut. Businesses must deal daily with employee records, IT infrastructure expenditures, sales information, and budgets. Individuals are spending more and more time reading texts, tweets, email, and blogs and viewing digital photos and videos. At a 2010 conference, Eric Schmidt, former Google chairman and CEO, noted, "every two days, we create as much data as we did from the dawn of civilization up until 2003."¹ Indeed, email has reached such a volume that Google recently added a Priority Inbox feature to its Gmail, which tracks what emails the user responds to and then uses predictive analysis to separate new mail into priority and normal lists.

This data deluge presents a business opportunity for determined organizations. Businesses that can extract useful patterns from data could secure a market advantage, lower their operating costs, and engage more of their employees. But the opportunity comes with an obstacle: Current data manipulation, extraction, and analysis tools fall short of translating data

into meaningful information. Wise action does not come from terabytes of data in a database; the records must be transformed into knowledge to add value to an individual, an organization, or a country. Umar Farouk Abdulmutallab is a case in point. Authorities had enough data to put Abdulmutallab on their radar before he attempted to bomb Northwest Airlines Flight 253 on Christmas Day 2009. But data did not translate to the information and knowledge they needed to prevent his boarding.²

One major problem is data scale. Microsoft Excel is one of the most popular data extraction, manipulation, and analysis programs—the tool of choice for as many as 500 million worldwide, according to Microsoft³—but teasing out information using spreadsheets or databases is tedious. Although Excel works extremely well for individual analyses, it quickly reaches its limit with larger data sets or when the user must explore multiple data facets simultaneously. In addition, Excel and similar programs keep data in a single place, making it difficult to share any analysis.

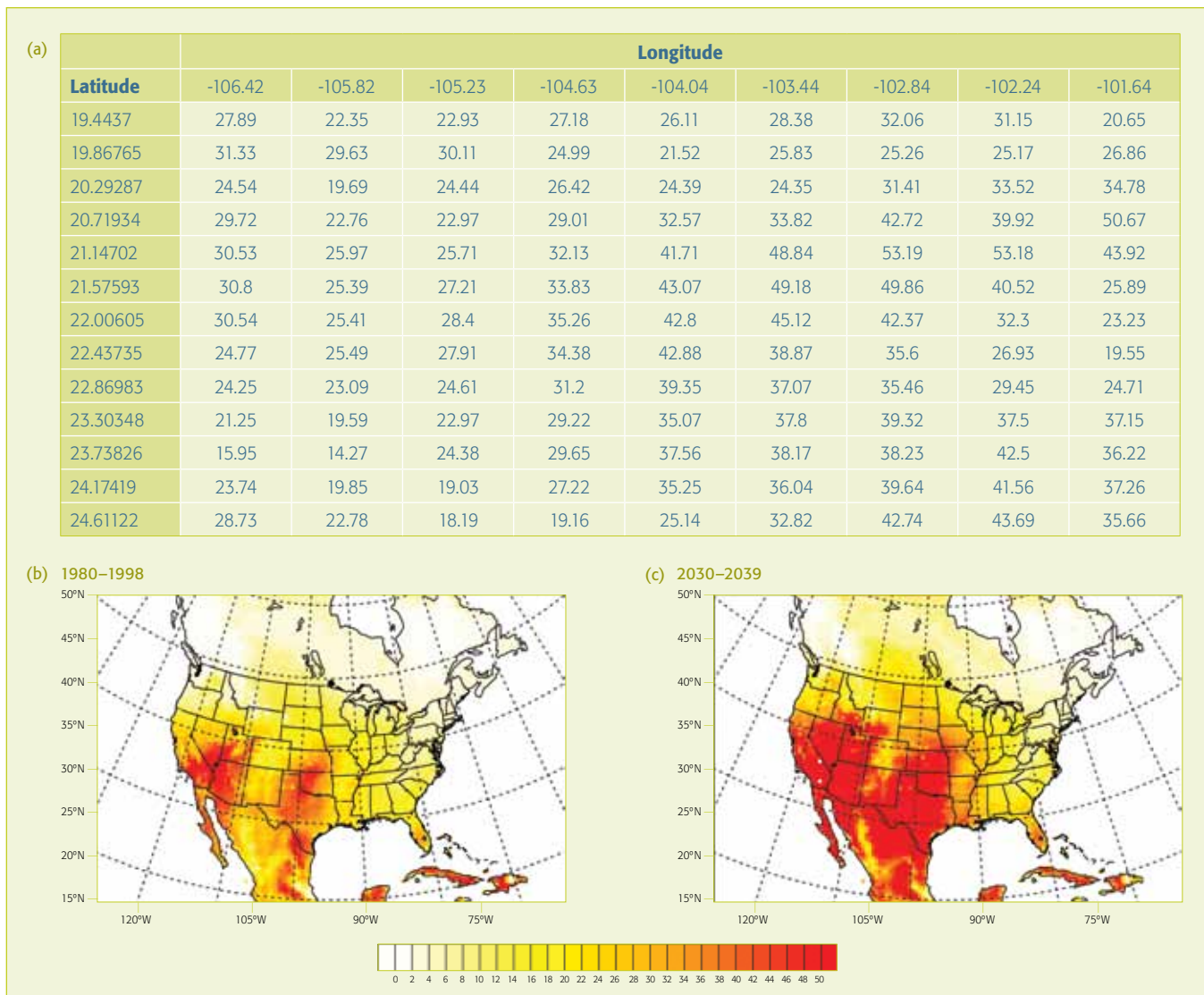
The good news is that tools for consuming and teasing knowledge from large data sets are evolving, with interactive data visualization through the Web being the first big step. As these tools mature, visualization will become the default data interface. Daily life will be more understandable as users become able to visualize their data trails in real time. Life dashboards will provide insight into how a meal in progress will affect energy and productivity in a few hours. Through organizational dashboards, managers will be able to evaluate policy decisions in real time, as employee data bubbles up. This enhanced interface will provide understanding into data at any time and place. And the building blocks for this ideal interface are already here.

The case for visualization

To make patterns discernible, a data interface must be as engaging as the user's favorite app or website and must facilitate

Inside Track

- Data visualization and Internet delivery form a powerful partnership: Visualization creates readily discernible patterns from raw data. Web delivery and social media platforms spawn discussion and collective understanding.
- Users demand tailoring from their applications and the websites they frequent. The interface and user experience for data analysis should be no less responsive.
- The largest hurdle to creating data visualizations is messy data, which must be cleaned, organized, and reformatted so that visualization applications can access it. New tools can help.
- Exposing data and applying usability best practices are two actions that organizations can take immediately as part of implementing visualization techniques.



data analysis for an entire group so that insights and ideas from analysis can propagate and spread innovation. In its simplest form, that interface might be a chart, graph, or map.

Regardless of its form, any such interface displays data as an image instead of as raw values, which encourages user interaction. The human eye can process images much faster than it can take in natural language, which is why people can compare lines, distinguish colors, and see patterns in location and color but not in raw numbers or text. Looking at data elements individually cannot create a comprehensive picture, particularly if the data is multivariate. A visualization interface exploits human cognitive ability, providing a way to extract knowledge from oceans of data.

Noblis is discovering the benefits of such an interface in two of its ongoing independent research projects. One effort is to downscale climate models to better understand climate changes at a local level. The other project relates commuting distance to a government employee's carbon footprint.

Climate data

Climate data models generate immense amounts of data. Without data visualization, Noblis analysts recognized that it would be practically infeasible to use the data to plan for climate change. How would rows of numbers reveal which nuclear power plants will struggle to maintain a proper cooling level as temperature rises, or which hospitals will be in the greatest

danger of fire during summer months? The best way to answer these questions is to view maps of temperature changes and increased fire risk along with overlays of hospitals and nuclear power plants. Answers would not come easily from lists of temperatures at various latitudes and longitudes. Figure 1a shows just a small part of the volumes of temperature data that analysts must sift through to discern any pattern that would help identify fire risk.

Such tables are a sharp contrast to the visualization in Figure 1b, which shows the increased risk of fire over time, given not only temperature, but also relative humidity, windspeed, and rainfall. Noblis analysts derived the visualization from volumes of risk data from the National Center for Atmospheric Research from 1980 to 1998 and then used Noblis' climate downscaling model to derive the visualization of projected risk in Figure 1c. Such projections can serve as a starting point for addressing risk-mitigation strategies.

Telecommuting policy

How does an employee's commuting distance relate to corporate telecommuting policy and how does this policy affect an organization's carbon footprint? These questions relate to multiple data facets, making them difficult to answer with data alone, particularly in a large organization. However, data visualization, such as the dashboard in Figure 2, makes it easier to coalesce and explore these seemingly disparate ideas.

The dashboard is based on the General Services Administration's Carbon Footprint and Green Procurement Tool Scope 3 Commuter Survey methodology, which Noblis developed for the General Services Administration to relate each government employee's commuting habits to his or her carbon footprint. The tool calculates information about the carbon footprint for each commuter. Noblis created the dashboard visualization using test data to showcase the power of further data drill-down and additional visualization methods. Users can interact with the dashboard to discern how individual segments of commuter behavior, such as commuting at a certain time, would impact the corporate carbon footprint.

The telecommuting example underlines the power of interfacing with data visualizations. Leveraging human cognitive ability and providing interaction among variables are two strong reasons to use a visualization interface. Because policy decision makers can quickly grasp interaction patterns among six variables, they have more insights into how they can align corporate telecommuting policy with sustainable business practices. They could not have gained such insight just by looking at raw datasets.

Visualization through the Web

Using the Internet to deliver visualizations and associated analysis techniques can amplify their benefit. Just as Web

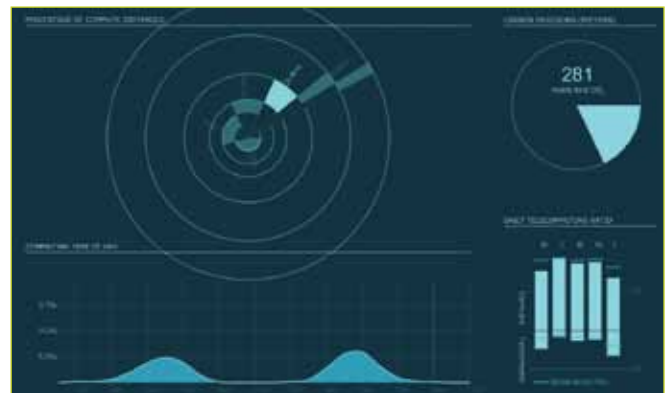


Figure 2. Noblis' dashboard visualization of sample commuter survey response data for employees that live 25 to 40 miles away from their workplace. The rose chart (top left) displays the percentage of employees by commuting distance. The pie chart (top right) displays the carbon footprint for the site. The bar chart (bottom right) displays the percentage of commuters versus telecommuters by day of the week. The graph (bottom left) filters the bar chart by driving hours. Clicking on a petal in the rose chart filters the data for a particular commuter subset.

2.0 wove user collaboration into the Internet's core fabric, so the amazing potential for sharing information and generating innovative ideas can be brought to bear on web-based visualization. The ability to comment on, interact with, and share ideas and information greatly increases the effectiveness of exploring visualized data. Internet and visualization form a powerful partnership: Visualization creates readily discernible patterns from raw data. Display through the Internet along with forums, blogs, or other forms of social media on the Web spawns discussion and collective understanding far beyond what is possible with visualization alone.

Tailoring

Delivering visualization on the Internet provides an additional benefit: User data and device characteristics can now inform the visualization to enhance the user's experience. If the user's device provides location data and the data being visualized is location based, the visualization can highlight the data closest to the user, offering a hyperfocused view. If the user is at a wide screen monitor or TV, the visualization can exploit the extra screen space to display more information. Likewise, if the user is on a smartphone, the visualization can be more focused to fit the constraints of the smaller screen. Such focused visualization can enhance interaction with zooming to provide more or less data, according to the user's needs and the device's touch features.

Users demand this kind of responsive experience from their applications and the websites they frequent. The data analysis interface and user experience should be no less tailored. Indeed, interfaces are rapidly entering novel dimensions of user tailoring. The sidebar "Visualization and Natural Interfaces" on p. 38 describes how natural user interfaces could be the ultimate in new models for tailored data visualization.

Handling Messy Data

The problem of messy data will likely be the most difficult problem to solve when moving to an enhanced data interface, but two new tools can point the way to a solution. Both can be used to great effect in working with messy data and bode well for future data manipulation tools.

Google Refine

Commenting on Refine's release in May 2010, Google calls the tool "a power tool for working with messy data, cleaning it up, transforming it from one format into another, extending it with web services, and linking it to databases" (<http://code.google.com/p/google-refine/>). With Refine, users can format data for integrity and layout consistency and then export it to the appropriate tool for storage and visualization. Essentially, Refine is an Excel-like program with a built-in housekeeper.

Data Wrangler

Stanford Visualization Group's Data Wrangler (<http://vis.stanford.edu/wrangler/>), currently in alpha release, provides a refreshing data transformation interface. Users import messy data, perform transformations using an intuitive graphical interface, and save results to any one of many formats. What makes Data Wrangler a standout is its ability to export user-made changes as a script. With this feature, a user can take several representative rows of a messy data set, make the necessary transformations, export the resulting script, and run the script against the much larger dataset. This feature greatly reduces the frustration of working with a medium to large dataset on the Web.

Open-source tools

Providing visually represented information when and where a user wants it creates a much better user experience. Open-source tools can help with the Internet delivery of both simple and complicated visualizations. Creators of simple visualizations, such as those using forms typical of Excel, can use several open-source libraries and free tools, including Flot, Google Chart Api, Raphael.js, and Highcharts. These libraries provide an excellent entry point to web-based visualization.

For more complex visualizations, Processing and D3.js allow for relatively easy web delivery.

Processing. Processing is part of a programming language and environment created specifically for data visualization. John Resig recently ported it to JavaScript, allowing Processing visualizations to be published to the Web plug-in free. ProcessingJs combines Processing with the newest parts of the Web; HTML5 and the Canvas element. Using Processing and ProcessingJs, developers can provide complex data visualizations in the Internet's natural language, which means that visualizations are available to users of any Internet-connected device—desktops, tablets, and smartphones.

D3.js. D3.js is a declarative JavaScript toolkit that makes it less tedious to create visualizations. Developers simply point data arrays or output from web-based application programming interfaces at the toolkit and then chain declarative functions on the data to output the chart or visualization. Because D3.js is

simply a JavaScript toolkit, anyone familiar with web technologies can use it and add interactive elements from traditional web applications to visualizations to enhance communication.

Implementation difficulties

Although web-based data visualization is far from conceptual, several implementation hurdles must be overcome before wide-scale adoption is possible.

Steep learning curve

Most toolkits for developing data visualizations require the knowledge of a programmer or a statistician. Modern toolkits are more powerful and easier to use than their predecessors, but they still have a learning curve, which can be steep for those with no programming knowledge. Organizations must consider such a curve in any investment.

Already, this implementation hurdle is becoming less daunting as software continues to advance. Tableau, for example, is an off-the-shelf tool that provides analysts with an easy-to-use interface for generating web-accessible dashboards and data graphics. There is no need for programming knowledge to create charts and graphs. Unlike Processing and D3.js, Tableau is confined to specific chart and graph types. However, successive tools should continue to expand Tableau's application scope and eventually overcome this limitation.

Modern toolkits are more powerful and easier to use than their predecessors, but they still have a learning curve, which can be steep for those with no programming knowledge.

Messy data

The largest hurdle to creating and using data visualizations is messy data, which must be cleaned, organized, and reformatted so that visualization applications and toolkits can access it. Whether they are HTML pages, health records, or employee satisfaction surveys, datasets tend to be disorganized. Even data that has some organized format tends to change over time because organizational requirements change, and data tracking requirements must adjust to meet this change. To create a comprehensive data picture, data must be consistent, which means it must be cleaned and formatted. The sidebar "Handling Messy Data" describes tools for these tasks.

Consistency is critical to the performance of aggregate operations, such as summation, average, and standard deviation, which in turn are crucial in creating visualizations. Without consistent data points, visualization will not be possible or, at the very least, the analysis and subsequently the visualization will have irregularities. Data cleaning and formatting are two ways to handle inconsistent data.

Cleaning. Incorrect or poor formatting occurs with almost any user-supplied data. The user might not have validated the data when submitting it through a form, there might be inconsistencies in local languages or terminology, or the problems could stem from something as simple as a typographical error. Although these errors seem inconsequential, they can lead to inaccurate results in data analysis and visualization.

Cleaning is a prerequisite for usable data. If possible, an organization should enforce desired formats when collecting data. If enforcement is impractical, the next best strategy is to set up rules for handling the most common issues. Not all data is created equal, but when data analysis is anticipated, that data must be clean before it is used.

Transformation. Data is found in almost any format, celebrity emails and tweets, X-rays, or point-of-sale information. To provide a better interface for this data, it needs to be in a format that computers can consume, which requires transformation.

Quirks with older browsers

Older browsers lack a complete implementation of current and forthcoming HTML standards, which means that developers lose the advantage of having a single code base for multiple devices. This in turn raises the challenge of developing layouts

for multiple devices, as “Responsive Web Design: Enriching the User Experience” on p.13 describes.

As researchers learn more about how users interact with mobile devices and tablets, layout design should become easier. However, at present, processors on mobile devices are typically weaker than on desktops and laptops, which means that mobile devices cannot render data visualizations as easily, especially visualizations that involve large data quantities or that rely heavily on animation.

Improving data interfaces now

Improving an organizational data interface can provide transparency in existing datasets and enhance their value. Exposing data and applying usability best practices are two actions that organizations can take immediately.

Expose data

Most companies have plenty of data, typically in isolated systems and in different formats, rarely exposed across organizational silos and hierarchies. If web-based visualization is to be the organization’s default interface, the underlying data sets must be available for use and reuse.

An excellent example of data exposure is data.gov, which hosts 389,681 datasets covering “every aspect” of government operations, from healthcare to public safety.⁴ These publicly available datasets have in turn spawned the creation of 236 applications—knowledge that didn’t exist before data.gov exposed its datasets.

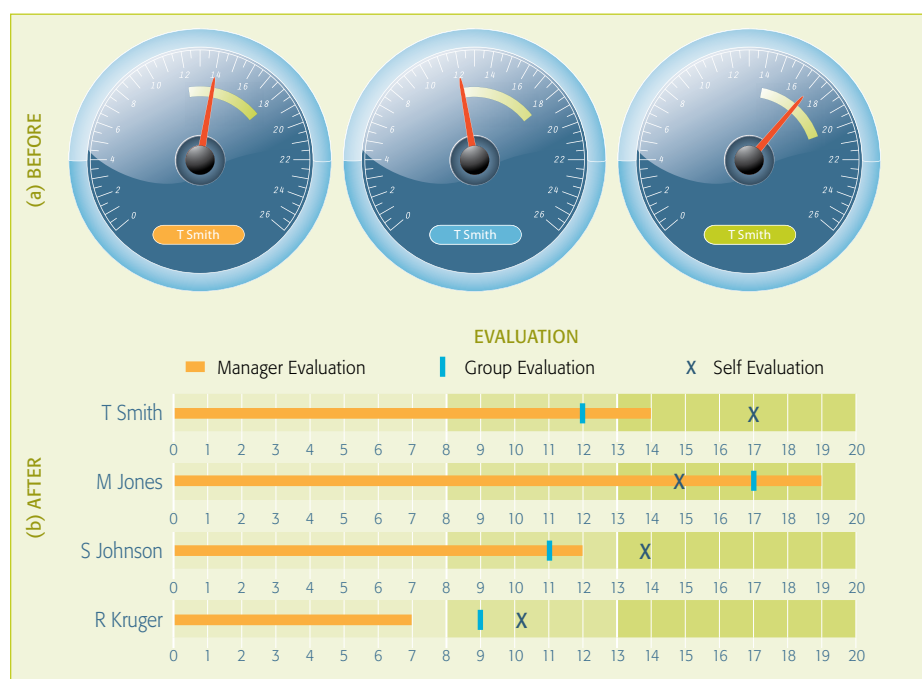


Figure 3. Evaluation data plotted in (a) gauges and (b) bullet graphs. The bullet graphs provide the same information in a much more readable manner, even when information is concentrated. (Bullet graph figure used with permission from S. Few, “Our Irresistible Fascination with All Things Circular,” *Visual Business Intelligence Newsletter*, Perceptual Edge Visual, March–May 2010.)

Visualization and Natural Interfaces

Just as interfaces moved from the clumsy command line to the graphical user interface, so they are taking another leap from screen-bound graphics to natural human action—gestures, voice, touch, and so on. Touch, in particular, is pervasive, with touchscreen devices driving mobile services. A survey by CEM4Mobile Metrics,¹ which looked at daily traffic in Internet mobile services in Finland, Sweden, Denmark, and Norway, showed a clear upward trend in the mobile touch market. From July 2010 to February 2011, the total market share for touchscreen devices had gone from 35.9 percent to 46 percent—a growth of 10.1 percent in just seven months.

In 2008, Microsoft released Surface, a platform based on Windows 7, which lets users interact on the same device through a 360-degree natural user interface (NUI) that supports touch and recognizes real-world objects without a camera. The platform leverages PixelSense technology that can support up to 50 simultaneous touches, reacting to fingers, hands, and objects as they are placed on the screen.

Emerging NUIs, such as Surface, offer an enhanced platform for leveraging data visualization. Collaboration on a single surface using natural interaction around visualization can enhance data analysis. An example related to epidemiological data analysis is the Towards Visual Analytics interface,

which allows data exploration through visualization aided by a multitouch display.² Using data visualization on these new interfaces, multiple users can directly interact with the data simultaneously. The result is a much higher level of collaboration than is possible using traditional interfaces.

Data visualization through NUIs can also change habits or behavior, such as energy use. Imagine digital signage in your home or office displaying real-time building performance such as energy consumption, air quality, occupant information, and operations and maintenance functions through advanced smart building features. Advancements like this coupled with touch-enabled operations are set to increase efficiency, transparency, communication, and collaboration in the near future.

References

1. PR Newswire, *New Report Shows Touch Screen Mobile Devices Driving Use of Mobile Services*, Apr. 11, 2011; www.prnewswire.com/news-releases/new-report-shows-touch-screen-mobile-devices-driving-use-of-mobile-services-119588939.html.
2. S. Flöring and T. Hesselmann, "TaP: Towards Visual Analytics on Interactive Surfaces," *Proc. Collaborative Visualization on Interactive Surfaces (CoVIS 09)*, University of Munich, Media Informatics Group, pp. 9–12, 2010.

Organizations can follow data.gov's lead by making corporate data appropriately available to their employees. Once given access to this data in a set location, individuals can manipulate the data to suit their purpose; visualize it, and share the results with others. Transparent data access can drive innovation and knowledge creation. As an organization demonstrates a commitment to openness and sharing, it sets the same expectation for employees, which can lead to collaboration that adds corporate value.

Apply usability best practices

For organizations to get the most from this kind of interface, data visualization must provide the answer being sought in a simple and appropriate form. It is easy to get carried away with design for design's sake. Gauges are a good example. Although they are prevalent in modern plug-and-play dashboard software, they are not always easy to read, compare, or analyze. In large part, this is because business executives are lured by their graphical appeal and the physical metaphor of a gauge. However, any visualization form must fit with the goal of pulling knowledge from data.

Stephen Few, author of several books on data visualization as it relates to cognition, notes problems with area-based charts and gauges. He writes, "By their very nature, circular gauges use a great deal of space to say relatively little, and they fail spectacularly when intended for comparison."⁵ The gauges in Figure 3 are not nearly as effective as the single bullet graph, which is simpler, more cohesive, and thus far easier to understand. Applying such human-centered design techniques accounts for visualization's goal of answering the question at hand and ensures that an organization is using visualization to its full potential.

Successful organizations understand their current situation and take action to enhance their position. Data analysis is an integral part of that process, and visualization is the new catalyst. To turn data overload into a data windfall, organizations need a better interface—one that recognizes how users think and offers a total user experience in data interaction. With web-based data visualization, organizations can provide data when and where the user needs it and tailor that data for specific user requirements. The result is rapid sharing, knowledge creation, engaged users, and more opportunities for innovation—all worthy goals for any business plan. ■

References

1. M. Siegler, "Eric Schmidt: Every 2 Days We Create As Much Information As We Did Up to 2003," *TechCrunch*; <http://techcrunch.com/2010/08/04/schmidt-data/>.
2. R. Goldman and H. Khan, *Timeline of Terror: Clues in Bomber Umar Farouk Abdulmutallab's Past*, ABC News, Dec. 30, 2009; <http://abcnews.go.com/US/timeline-terror-clues-bomber-umar-farouk-abdulmutallabs-past/story?id=9449255>.
3. *Turn 500 Million Excel Users to Your Deployment Opportunity: Parallelism with HPC Services for Excel*, Microsoft Showcase, July 9, 2010; www.microsoft.com/showcase/en/us/details/8750ad1d-b854-4b73-aa95-6946fc6a9f9a.
4. V. Kundra, *From Data to Apps: Putting Government Information to Work for You*, The White House Blog, May 20, 2011; www.whitehouse.gov/blog/2011/05/20/data-apps-putting-government-information-work-you.
5. S. Few, "Our Irresistible Fascination with All Things Circular," *Perceptual Edge Visual Business Intelligence Newsletter*, Mar./Apr./May 2010; www.perceptualedge.com/articles/visual_business_intelligence/our_fascination_with_all_things_circular.pdf.



Seth A. Blanchard is a senior staff member at Noblis, where his experience includes web application development. He received a BS in applied computational mathematics from Virginia Polytechnic Institute and State University. Contact him at [seth.blanchard@noblis.org](mailto:blanchard@noblis.org).



No More Black-box Disaster Response

Matthew B. Holtry and Judy A. Vo

Social networks lay the foundation for “crowd sourcing,” which leverages the tweets, texts, and messages that people post during and after a disaster. The challenge is to channel that information through effective website design so that first responders continue to get accurate and timely data.

In February 2011, when Libyan citizens rebelled against Muammar el-Qaddafi and violent retaliations were launched, the United Nations Office for the Coordination of Humanitarian Affairs moved swiftly to respond to the crisis. At first, it was unable to act because no information management officers were stationed in Libya, and the government-run media was not trustworthy.¹ Despite this handicap, in 48 hours, it had in hand more than 100 compiled response activities—data that took four weeks to amass in the Philippines, two weeks in Haiti, and two weeks in Pakistan after their respective crises.²

This invaluable information came from volunteer crisis mappers using information communication technologies linked to social media platforms. Such groups typify the *crowd sourcing* phenomenon that stems from the increasingly broader application of social networking. In the aftermath of natural and other disasters, crowds become self-organizing problem solvers, often broadcasting geocoded tweets for help translated into multiple languages and plotted on maps in near real time. On the basis of this input, locals assess the situation on the ground, corroborate reports, and provide triage for response personnel thousands of miles away. Often these efforts provide

a comprehensive portrait of the situation, alerting response personnel of road closures or hazardous situations. Even more astounding is that all this coordination happens spontaneously at virtually no cost to governments.

But crowd sourcing is not without risk. Information could be distorted or filled with extraneous details that can cripple decision making or lead to solutions that harm instead of help. Unscrupulous individuals can pose as disaster-relief charities in an attempt to swindle survivors, similar to what happened after the March 2011 earthquake in Japan.³

To mitigate or even eliminate these dangers, organizations can implement a social interaction design strategy that is based on research in social convergence theory and user experience design. Elements of such a strategy exploit the desires and motivations of actors within the context of a disaster situation and lead to crowd-sourcing solutions that generate useful, actionable data for responders.

For some disasters, such as severe weather events, crowd-sourcing solutions might be the only viable strategy in the near future. Such events are intensifying and occurring more frequently. As this trend continues, the traditional top-down model of disaster relief will be overwhelmed, and the current fiscally constrained environment will make it impractical to add top-down resources. This combination of increased potential for natural disasters and decreased ability to fund a response strongly implies the need for a bottom-up model like crowd sourcing.

Inside Track

- During and after a disaster, crowds become self-organizing problem solvers, often broadcasting geocoded tweets for help translated into multiple languages and plotted on maps in near real time.
- Motivation has long been a topic in social convergence research, and the results of these studies can greatly aid disaster-response programs in understanding the motivations of actors in a disaster.
- The Facebook group, “I’m ok at VT” had over 2,000 members in the first 24 hours of its creation. The American Red Cross’ “Safe and Well” website had only 4,092 records over nine months in response to three significant crises.
- Five key guidelines can help a website encourage larger participation and bring more resources into the crisis-mapping dialogue.

Self-organized decision-making

In any effective crowd-sourcing solution, the crowd must have the authority to make decisions and receive feedback about how their actions led to an outcome. Without these traits, the crowd has no motivation for participation.

Unfortunately, this philosophy often collides with traditional command and control approaches. Consider a request for a flash mob—a spontaneous gathering in which individuals gather,

Table 1. Characteristics of seven social convergence types and associated risk of hindering triage efforts.

Convergent Behavior Type	Motivation and Contribution
The Helpers	Assist victims or disaster response efforts
The Anxious	Identify loved ones who were impacted by the event
The Supporters	Provide encouragement and gratitude for response efforts
The Curious	View the disaster impact
The Returnees	Assess losses and salvage property
The Mourners	Mourn the disaster victims
The Exploiters	Profit personally from the disaster

 No Risk; High Contribution  High Risk; Low Contribution

act to fulfill a single purpose, and then quickly disperse. In one experiment at Juniata College in Huntingdon, Pennsylvania, a researcher attempted to initiate a flash mob by sending a text message to 100 students that read, “Meet@Ellis Ballroom 2nite@10:30. Big event. Tell others.” The “big event” was not defined in the message, but it consisted of sugary snacks and a brief survey to collect research data. At 10:30, the gathering had only one participant—the researcher who originated the message.

Why did the flash mob experiment fail? The text went to a group that could anticipate a shared interest (being a research participant), but the message did not sufficiently motivate the members to “tell others.” They were not working collaboratively to solve a problem, and it was not obvious what they would receive for participating. From a command-and-control design view, the experiment provided no feedback, few rewards, and a lot of uncertainty.

These results are not atypical in traditional disaster relief. Indeed, communication and motivation problems have sparked the development of an entire field of study—crisis informatics—dedicated to how people interact and use technology during and immediately following disaster events.⁴ Dr. Leysia Palen, an expert in this field recently concluded that increasing amounts of federal spending are aimed at the development of organizational processes and technology for disaster management, which is often “detached from the century-long work of empirical study on the topic.”⁵ In other words, governments might be paying a lot to develop tools that don’t work.

In contrast, disaster-relief solutions based on crowd sourcing have proven highly effective in meeting government needs in a variety of situations. The sidebar “Crowd Sourcing for Victim Identification” on p. 42 describes the success of this approach after the shootings at the Virginia Polytechnic Institute and State University.

According to the Federal Emergency Management Agency’s (FEMA’s) *National Response Framework*, data collection, tracking, and incident reporting is to support decision making and to rapidly summarize and prioritize information. The report noted that information must be gathered accurately at the scene and effectively communicated to those who need it and that “a

common operating picture is essential.”⁶ Social media platforms combined with omnipresent information and communication technology have allowed self-organized, all volunteer groups to perfectly fulfill each of these FEMA criteria.

Understanding motivation

If these crowd-sourced disaster relief solutions are so effective in some instances, why do they appear to fail in others? As the account of the flash mob experiment illustrated, failure often stems from a lack of motivation within the situational context.

Motivation has long been a topic in social convergence research, and the results of these studies can greatly aid disaster-response programs in understanding the motivations of actors during and after disaster events. Such research also has principles for applying user experience design to attract groups who will add value to the relief effort.

When disaster events occur, individuals with a wide range of motivations converge on the disaster site. In an increasingly connected world, social convergence has become virtual, obviating the need to rely on proximity to the disaster site. For example, Ushahidi’s technical team launched a site to collect information about victims of the 2010 Haiti earthquake within hours of the event.⁷ They then virtually trained 300 volunteers from Portland to Washington, DC, to Montreal to Geneva to London, including some members of the Haitian Diaspora, to continue mapping thousands of messages around the clock from Haitians living through the experience. FEMA and the Marine Corps responders in Washington, DC, and Haiti relied on the resulting map to tactically deploy resources.⁷ With virtual social convergence, more actors with larger social networks and a wider range of expertise can engage in collective problem-solving to support disaster relief.

Behavioral types

Social behavior research has traditionally identified seven types of social convergence that occur during and immediately following a disaster event. Each of the groups has its own motivation ranging from The Anxious who want to identify loved ones who were impacted by the event to the Exploiters who want to personally profit from the disaster. Table 1 shows seven behavior types and what motivates individuals with this behavior.⁸

Some behavior types—Helpers and the Anxious—can assist relief efforts, since their motivations are aligned with those of disaster responders. Other behavior types may hinder relief efforts by adding too much noise to the situation. Twitter messages from Supporters, such as “Our prayers are with the victims,” might boost morale, but they do not assist with victim identification or disaster triage. Imagine the confusion and wasted time for first responders if these messages were geocoded and plotted on a map next to “I’m trapped in the basement” tweets.

The importance of Helpers

Clearly, effective disaster relief based on crowd sourcing must apply user experience design to attract the desirable behavior types while discouraging participation from the unwanted ones. The most desirable type is the Helpers. By creating a user persona for Helpers, such as that in Figure 1, organizations such as aid agencies can more effectively design a solution that attracts these individuals.


A user persona is an archetype that explains the user's demographics, cultural and social considerations, technical comfort, and motivations that drive user behavior. For example, Helpers are altruistic, with a strong desire to feel as if they're making a contribution that would not exist without their input. As such, Helpers need little direction, since they're often experts in their respective fields, but they require feedback about how their actions are contributing to crisis relief. They want to know that they are needed and that they are making a difference.

This final trait may explain why formal command and control sites with a central authority, such as FEMA and the American Red Cross, tend to be less popular with Helpers than informal grassroots movements, where everyone becomes a citizen journalist. On an informal site, anyone with Internet access can contribute to the dialogue, plot data on a map, and influence crisis response resources.

In contrast, a FEMA-created crowd-sourcing site with a prominently displayed Department of Homeland Security logo tends to put off Helpers, who might question the value of their contribution. They might assume FEMA has all the technology, knowledge, and resources that it needs to respond to the crisis, and that the Helpers' skills are not needed. Consider this revealing example of participatory difference in informal and institutional sites. The Facebook group, "I'm ok at VT" counted over 2,000 members in the first 24 hours of its creation in response to a single significant crisis. The American Red Cross' "Safe and Well" website had only 4,092 records over nine months in response to three significant crises (all in the United States).⁹

Design for the Helper persona

The Helper persona assists victims or responders by providing information about personal and property safety and sources of relief.⁸ To encourage Helper participation in disaster relief, a crowd-sourcing tool should empower Helpers to improvise important activities, such as mapping the situation on the ground. Consider a scenario in which survivors are trapped in a house on a road blocked with debris. Distributed Helpers who are self-coordinating the mapping on the basis of messages from actual survivors will produce a map far more quickly and efficiently than outside first-responders, who can map only the areas they can physically access. Organizations can then use the resulting map to deploy resources to areas with the most need and communicate progress back to the Helpers.



Brian
Persona Type: Helper
Age: mid-30's
Location: Suburbs near large city
Technical Comfort: Savvy
Job Title: Web developer

Back Story

- Studied computer science at a public university.
- Formerly worked for a large corporation, but felt he had more to contribute.
- Returned to school for an advanced degree and now works for a non-profit.

Motivations

- Restoring social equity through charity work and providing ad-hoc assistance to others.
- Use his passion for open source software to build maps that help more people solve problems.

Frustrations

- Lots of people need help and lots of people want to help, but it can be difficult to connect the two groups to each other.
- Charities do not always coordinate efforts together which can waste resources.

Their Ideal Experience

- Seeing the direct results of his contributions.
- Receiving immediate feedback.
- Being considered a team member.

Figure 1. Profile of a Helper persona. By categorizing and defining people's traits, an organization can better identify what motivates an individual and what they can contribute to crisis response.

Noblis has evolved five key guidelines for designing a website to promote crowd-sourcing disaster relief that attracts and motivates Helper participation. By incorporating these user interaction design elements, the website is likely to encourage larger participation and bring more resources into the crisis-mapping dialogue.

Eliminate all contribution barriers

The website should remove any obstacles between the user and the user's ability to contribute content to the site. Because the Helper is motivated by a desire to contribute information and expertise, the page should use text sparingly, substituting graphics and whitespace where possible to allow for faster interpretation of site navigation. The homepage should minimize the number of clickable paths to prevent users from feeling paralyzed with too many options. Most important, the site must not require a long registration or verification process—the longer Helpers must wait to contribute content, the more likely they will be to seek another channel.

Demonstrate a need for citizen input

To encourage Helper involvement in problem solving, the site should downplay official logos of large relief organizations and increase the prominence of postings from other Helpers. One strategy is to post a live Twitter feed; another is to display a running list of recent posts on the crisis map. Both strategies play to the Helper's need to feel valued.

Crowd Sourcing for Victim Identification

This account is taken from “Crisis Informatics: Studying Crisis in a Networked World,” (L. Palen et al., University of Colorado; www.nccss.ac.uk/events/conference/2007/papers/paper172.pdf).

Within 17 hours of the first shooting at Virginia Tech, self-organized groups using Facebook and Wikipedia had successfully identified 14 of the 32 victims. By the time the on-site officials released the official report the next evening, an accurate and complete list of the victims had already been established, verified, and posted online by an international group of people who engaged in a somber dialogue to answer one key question: Once they assembled all of the text messages, online communications, and other signs of physical activity throughout their collective social network, who was missing? By leveraging their combined access to information, the crowd quickly and effectively answered its question in this crisis situation without any central authority to guide the process.



Photo from UPI/Roger L. Wollenberg; www.upi.com/enl-win/5fd017e6c9c65c4c04b930221411375a/.

Streamline dialogue

Helper posts must be displayed in near real-time to build an immediate connection between the user and the unfolding dialogue. In a crisis situation, Helpers cannot afford to have their content delayed or bottlenecked through hierarchical approval processes. By allowing citizen journalists to corroborate each other's postings and even assign a simple priority ranking (such as 1 to 10) to individual crisis events, site design can foster self-policing activity that requires no centralized hierarchy.

Provide feedback

Helpers need feedback to measure their personal impact. Site design should give individuals and large relief organizations alike the incentive to change the status of issues immediately as they are resolved and then communicate that resolution back to the author of the original posting. For example, if a Helper tweets about a person trapped in a building, he should receive a tweet, email, or other notification when that person has been freed. By closing this communication loop, the Helper's altruistic intentions are flattered, and he is encouraged to continue contributing.

Encourage gaming behavior

The site should reward super-Helpers—those who post the highest number of verified and resolved crisis events—with special recognition. Alternatively, the responders themselves, such as FEMA representatives or local police, could receive recognition according to the number of issues they resolve. After the immediate crisis has passed, traditional media and journalists might interview these super-Helpers or -responders, triggering viral marketing campaigns that encourage other Helpers to participate in future crowd-sourcing relief efforts.

A model for crowd-sourcing design

As described in a FEMA Emergency Management Institute course, convergers might be present precisely because formal responders are unable or unwilling to meet these needs.¹⁰ A case in point is Ushahidi.

Ushahidi, which is Swahili for “testimony” or “witness,” uses eyewitness reports from the public to map the location of violent post-election outbursts in Kenya. The nonprofit organization develops free, open-source mapping and data visualization software, providing many examples of how the new digital world promotes decentralized mapping of crisis information for the general public; the government; and most important, first responders.

Ushahidi was developed in 2008 by a team of open-source coders. Its main purpose, according to Ory Okolloh, one of Ushahidi's founders, is to gather more and better information about post-election violence, but the Ushahidi platform has exceeded its developers' goals; it is now used worldwide to collect and map information in real time on the basis of messages, email, tweets, and web-based applications.¹¹ From mapping lion and elephant sightings for a Kenyan wildlife charity to the most recent Japan earthquake and tsunami, the Ushahidi platform is front runner in emergency response.

Effective design elements

Figure 2 identifies the design elements of the Haiti Ushahidi site that led to a successful rallying of volunteers. Key elements include multichannel data-gathering technology, a How to Report box, and Crowd Map software.

The How to Report box instructs witnesses to submit information by messaging or texting and provides a local and international number, an email address, and a web form to submit directly on the site.¹² The data feed is then sent to a team of volunteers who reads these messages and plots the information on the map.

Crowd Map is a hosted version of the Ushahidi platform that enables visitors to set up a platform in two minutes.¹² Being able to quickly initiate crowd sourcing is useful when other communication networks rapidly become saturated or unavailable.

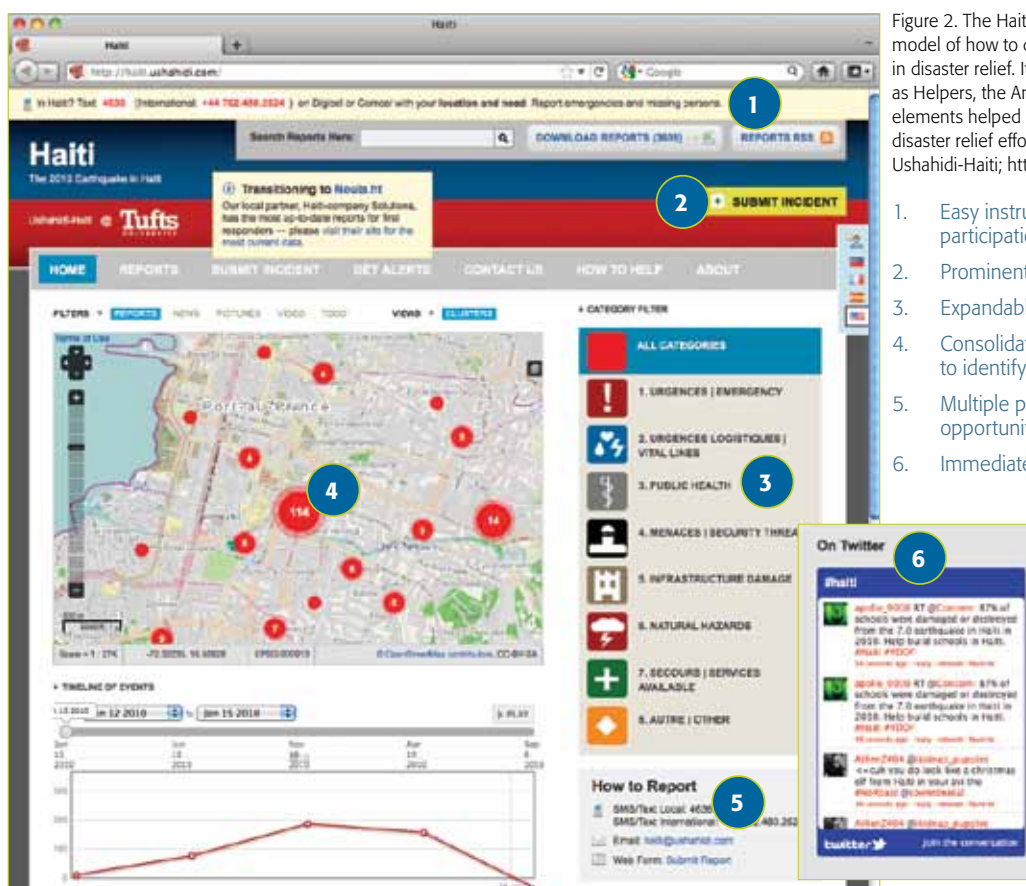


Figure 2. The Haiti Ushahidi website. The website is a model of how to design for effective use of crowd sourcing in disaster relief. It aims to attract useful contributors, such as Helpers, the Anxious, and the Curious. These design elements helped attract and engage volunteers in the Haiti disaster relief efforts. (Image used with permission from Ushahidi-Haiti; <http://haiti.ushahidi.com>.)

1. Easy instructions to promote quicker participation
2. Prominent display of most popular link
3. Expandable crisis categorization
4. Consolidated incident mapping to identify highly impacted areas
5. Multiple platforms expand engagement opportunities
6. Immediate display of posted content

Engaging of multiple personas

Much of the Ushahidi platform's success is attributable to the organization's understanding of online social convergence during a disaster. The site design integrates features that attract and motivate Helpers, the Anxious, and the Curious—all social convergent types that positively contribute to disaster relief.

Ushahidi fosters the involvement of Helpers by supplying real-time information on where help is most needed. Updated Twitter feeds appear instantaneously on the page, reporting disaster and relief effort locations as well as pleas for help. During the Haiti earthquake, Haitian-Americans were saving lives by translating messages from Creole for search and rescue teams trying to locate survivors.¹³ The site also enables Helpers to get direct feedback from their contribution. Help Tag Photos and Person Finder are resources that attract both Helpers and the Anxious.

Another attraction for both Helpers and the Anxious is Ushahidi's endorsement of itself as a reliable source through standard verification processes for disaster relief based on crowd sourcing. The "Ushahidi Guide to Verification" provides several ways for users to verify information, and the site cautions visitors against poisoned data—data that is false or created to function as bait or create a diversion. Visitors are advised to check that the same information is coming from multiple

sources before verifying that the information is correct.¹³

The ability to toggle between verified and unverified data is a lure for the Curious, who tend to visit to watch the activities surrounding the disaster, and then stay to become Helpers when they see a need.

End-to-end design steps

With the current variety of technology, almost anyone can set up a social media site and begin adding information to the Web. The challenge is how to engage users enough that they continue contributing valuable data. By taking four steps, summarized in the sidebar "Implementation Steps" on p. 44, organizations can reap the benefits of crowd sourcing in a response solution.

Develop user personas

Identify key user groups and conduct user research by interviewing a cross-section of the intended user community or, if time is short, by using market data. Use the research findings to develop user personas, recognizing the needs, motivations, and goals of each. This activity helps lay the foundation for long-term relationships in which the user continues to return and use the site, even when help is no longer imminent.

Implementation Steps

Before beginning any project that applies social media to disaster response, an organization must first identify and understand the needs and desires of its users and then apply the principles of user-experience-driven design to create a response website. Following these four steps should lead to a successful project:

- Define key attributes of each social convergence type.
- Design a social interaction focus around a specific type.
- Implement gaming strategies to promote continued user interaction.
- Provide feedback loops.



Design social interaction

To create satisfaction from contributions and thus encourage continued social interaction, design and create interactions that are centered on the goals of each persona. Disaster-response sites often run at full capacity only during or immediately after a crisis. To encourage use at other times, include features like case studies and Q&A to let users share experience or expertise.

Implement a gaming strategy

A gaming strategy can be the difference between a system's successful adaptation and its failure. It's not enough to have the most visits or hits for a site; returning users are just as important as new ones. To determine the best game theory to apply, apply the game dynamic that appeals most to the needs and desires of a user persona. Create an emotional connection that fuels the ideal experience. "Toward a User-Centric Digital Ecosystem" on p. 4 describes some game dynamics and their goals.

Provide feedback

To facilitate verified data on disaster relief sites, provide feedback for individual contributions. Not only does this allow for immediate satisfaction of seeing the impact of their own contributions, the site is able to market itself as a reliable resource for help and information in the event of an emergency.

Several strategies are possible: let other users vote on how helpful the post is, raise the individual's status according to amount of contribution, or upgrade access/permission levels. The goal is to encourage users to continue contributing or checking in to see what other users think of them.

Break storms, terrorist attacks, and earthquakes bring chaos and unpredictability. In light of social networking's staying power, emergency response is no longer a call to 911. The world requires a new model for synthesizing data in the immediate aftermath of a disaster event and efficiently deploying resources where they can best be used. This activity must be achieved in a fiscally-constrained environment where it may be impossible for local, state, or federal governments to radically expand on-the-ground resources.

Crowd sourcing offers a no-cost solution for mapping and prioritizing individual crisis events after a disaster. Large aid organizations can leverage the power of self-coordinated problem-solvers to deploy their resources more effectively. By applying social convergence theory and user experience design to these solutions, crowd-sourcing sites can better exploit the desires and motivations of Helpers and other beneficial parties that contribute to saving lives, protecting property, and meeting basic human needs. ■

References

1. P. Meier, *Patrick Meier—Changing the World, One Map at a Time*, (uploaded by republica2010 on May 3, 2011); accessed June 4, 2011; www.youtube.com/watch?v=Hh_PIVqf8BA.
2. *Libya: How Online Mapping Helped Crisis Response*, AlertNet, May 12, 2011; www.trust.org/alertnet/news/libya-how-online-mapping-helped-crisis-response.
3. *Scam Alert: Fake Japan Disaster Relief Charities*, Oregon Department of Justice, Mar. 15, 2011; www.doj.state.or.us/releases/2011/rel031511.shtml.
4. University of Colorado, *Crisis Informatics*, June 2011; <http://disaster.cs.colorado.edu/>.
5. L. Palen et al., *Crisis Informatics: Studying Crisis in a Networked World*, Univ. of Colorado, Boulder, 2007; www.ncess.ac.uk/events/conference/2007/papers/paper172.pdf.
6. U.S. Dept. of Homeland Security, *National Response Framework*, Jan. 2008; www.fema.gov/pdf/emergency/nrf/nrf-core.pdf.
7. *Haiti: Taking Stock of How We Are Doing*, The Ushahidi Blog, Feb. 6, 2010; <http://blog.ushahidi.com/index.php/2010/02/06/ushahidi-how-we-are-doing>.
8. J. Kendra and T. Wachtendorf, "Reconsidering Convergence and Converger-Legitimacy in Response to the World Trade Center Disaster," *Research in Social Problems and Public Policy*, vol. 11, 2002, pp. 197–224.
9. A. Hughes et al., "'Site-Seeing' in Disaster: An Examination of On-Line Social Convergence," *Proc. 5th Int'l ISCRAM Conf.*, May 2008; <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.157.3104>.
10. FEMA Emergency Response Institute, *Catastrophe Readiness and Response Instructor Notes for Session No. 9*, 2011; <http://training.fema.gov/EMIWeb/edu/docs/crr/CAT%20-%20Session%209%20-%20Planning.doc>.
11. "Ushahidi: Crowdmapping Collective That Exposed Kenyan Election Killings," *Guardian News*, Apr. 2011; www.guardian.co.uk/news/blog/2011/apr/07/ushahidi-crowdmap-kenya-violence-hague.
12. *Ushahidi, The 2010 Earthquake in Haiti*, Tufts University, 2009; <http://haiti.ushahidi.com>.
13. "Ushahidi Guide to Verification," *Ushahidi Community Resource*; http://community.ushahidi.com/uploads/documents/c_Ushahidi-Verification-Guide.pdf.



Matthew B. Holtry is a lead staff member at Noblis, where his experience includes sustainability issues with emphasis on the management of greenhouse gas emissions. He received an MBA from Pennsylvania State University. Contact him at matthew.holtry@noblis.org.



Judy A. Vo is a staff member at Noblis, where her experience includes Scrum Agile Software Development. She received a BS in business information technology, decision support systems from Virginia Polytechnic Institute and State University. Contact her at judy.vo@noblis.org.



HTML5: What's Different for User Experience Design and the Web?

Daniel Y. Na and Brian C. DeRocher

HTML5 fixes many of HTML4's problems, emphasizing usability while capitalizing on Internet growth. Organizations that understand its features can better position themselves in the upcoming shift to a semantic web.

With an estimated 46.5 billion webpages indexed by Google alone by 2011,¹ there is seemingly no end in sight for web growth. From the creation of Hypertext Markup Language (HTML) in 1989 to the robust, multimedia-rich web applications of today, the Web's utility has increased exponentially in a short time. It has evolved from humble beginnings as a simple information exchange to an interactive and user-centric ecosystem, building on its users' desires to socialize and collaborate across the globe.

The technical capabilities of websites, browsers, and devices have kept pace with this growth. Website creators now draw on a variety of programming languages, libraries, and programming frameworks to deliver content. Static, graphically sparse webpages viewed solely through a desktop monitor have given way to interactive and innovative modern websites, which increasingly target mobile phone and tablet devices. Forward-thinking companies recognize the imperative to stay abreast of such rapid change.

Inside Track

- By including new tags, application programming interfaces, and semantic meaning, HTML5 seeks to meet the demand for increased function and interactivity within web applications.
- Developers can leverage the native capabilities of HTML5-compliant browsers, eliminating the need for many third-party code libraries and software. In this way, they can offer visitors a more consistent user experience across websites.
- HTML4 provides no native means to play video or audio, forcing users to rely on proprietary third-party software delivered through browser plug-ins. HTML5 offers the welcome relief of native video and audio support.
- The semantic markup tags within HTML5 enable search engines and programs to index webpages faster. Users get greater search efficiency, and organizations can disseminate information with greater precision.

As the newest HTML specification from the World Wide Web Consortium (W3C) HTML Working Group, HTML5 is an example of an emergent technology within web development. HTML5 is foremost an improvement over several functional limitations in HTML4-based websites. It includes new sets of tags, application programming interfaces (APIs) and tools to introduce usability and performance improvements across the Web.

But HTML5 is also a vehicle by which the W3C can introduce Web 3.0, or the semantic web. In response to the skyrocketing volume of Internet data, the semantic web aims to fundamentally shift webpage structure to facilitate the understanding of that data. A meaningful webpage structure will help search engines and programs better filter data across massive datasets, delivering information to web users faster and with greater precision.

Although the specification is not yet final, several HTML5 features are mature enough that organizations can begin to evaluate them and anticipate how HTML5 will enhance their products and services. At its core, the new standard seeks to improve the Web experience for all stakeholders—users, content providers, developers, organizations, or machines.

Addressing HTML4 limitations

The introduction of HTML4 in 1997 was a significant milestone for web development in part because it introduced the foundational components of the modern webpage: Cascading Style Sheets (CSS) for presentation, scripting languages, accessibility tools, tables, and media elements.² However, since HTML4's creation, the Web's technological landscape has changed significantly. More users are accessing sites through smartphones and tablets, which present obstacles foreign to desktop devices. Webpages must respond to smaller screen resolutions or lower performance while accommodating a range of hardware, input, and size restrictions.

To meet these and other needs, developers have created or located suitable solutions, including additional programming languages and software libraries—all of which require the installation of third-party software. Although these solutions work to some degree, they are in essence a patchwork, which can present serious disadvantages.

A major drawback is the *lack of standardization*. With the variety of available solutions, developers are apt to choose different libraries or software to accomplish the same task. For example, although web form validation appears almost universally across the Web, there is no standard solution, which puts the burden on developers to integrate their chosen solution with the rest of the website. Thus, a user who visits nine websites might encounter nine different validation forms.

Another disadvantage is *increased page load time* from the required extra software, creating sluggish page manipulations on devices with lower performance or limited connectivity.

Possibly the most serious disadvantage of patchwork solutions is an *inconsistent user experience*. When sites depend on the inclusion of specific libraries or software such as Adobe Flash, the user experience is inconsistent across platforms. For example, Flash support on mobile browsers is highly variable—dependent on device, browser, and application functionality. And although the latest Android iterations include Flash 10.3, browser support on mobile does not fully mirror what desktop device users can enjoy. As a case in point, as of September 2011, for all its videos accessed by either the Motorola Droid X and Motorola Atrix 4G running Android 2.3.3, the popular video site Hulu.com displays an error message, “This video is not available on your platform.”

Two noteworthy examples of HTML5 enhancements are web form validation and multimedia access support.

Web form validation

To protect against errant user input, HTML4-based websites implement common form validation through JavaScript code. HTML5 eliminates the need for such code.

In HTML4

E-commerce and web interactivity created a need for forms for everything from user community memberships to product orders. HTML4 supports a variety of input types for form creation—text fields, checkboxes, buttons, and so on—but it does not provide built-in validation to identify correct input and instruct users on how to complete a form correctly. For example, an input field marked “Telephone number” should restrict input to common telephone number formats (1231231234 or 123-123-1234). To have a positive experience filling out forms, users need a mechanism that provides feedback for invalid input.

In HTML4, web developers who want to provide more satisfying form validation must write their own validation code using languages like JavaScript, or they must use existing form validation libraries. Figure 1 shows how jQuery Validate, a third-party JavaScript library, identifies incorrect form input.

In addition, HTML4 web forms cannot provide intrinsic function to different input types—a limitation that impedes usability. For example, HTML4 alone cannot capture date values except through text input, such as “Enter the date in format MM/DD/YYYY.” This method is susceptible to entry error and can be frustrating to users who are accustomed to thinking of dates on a calendar or in some other text format, such as day before month.

Third-party JavaScript libraries let developers work around the problem by presenting a visual calendar on the form, as in Figure 2.

In HTML5

To eliminate these workarounds HTML5 provides a variety of specific input types, each of which contains intrinsic functionality. Table 1 displays some of these new tags; a full list is available at www.w3schools.com/html5/tag_input.asp.

Consequently, developers no longer need to include a JavaScript library for users to see a calendar. Figure 3 shows the date and color HTML5 input types.

HTML5-compliant browsers also provide form validation for HTML5 input elements. Figure 4 shows an example of validation that has detected an incorrect email address format.

By providing context-specific input types, HTML5 allows webpages to accommodate mobile users in specific ways. In Figure 5, a Safari user sees telephone-only inputs when entering a telephone number with the tel input type.³

Table 1. HTML5 input types and purpose.

Input Type <input type="">	Purpose
Search	Search boxes
Number	Spinboxes
Range	Sliders
Color	Color pickers
Tel	Telephone numbers
Url	Web addresses
Email	Email addresses
Date	Calendar date pickers
Time	Timestamps

Figure 1. Form validation with jQuery Validate. With HTML4, developers commonly rely on JavaScript for form validation

Figure 2. Date input in HTML4. (a) The user must input a date in a fixed, text format. (b) To enhance the user experience, the developer can add a JavaScript calendar through the jQuery-ui library.

Figure 3. HTML5 input types. Using the Opera browser, the user can select (a) a calendar or (b) a color rather than input text for each type of input. HTML5 provides these and other input types natively in the browser without requiring additional JavaScript.

Figure 4. Validating an email entry in HTML5 using Opera. All HTML5 input elements include validation built into the browser with no need for JavaScript.



The Bottom Line

HTML5's provision of controls natively within the browser serves both developers and users. Developers eliminate the extra time to implement third-party validation and form libraries. Users enjoy a faster page load and more consistent form functionality. Form validation can become standardized because there is no longer a need to adapt to each site's form validation choice.

The HTML5 logo is the property of the World Wide Web Consortium.

Multimedia access

One of the most popular uses of the Web is to access video and audio content. Projections indicate that video traffic alone will account for more than half of consumer Internet traffic by the end of 2012, becoming the single largest Internet traffic source.⁴ Although HTML4 requires third-party audio and video players to play multimedia content, HTML5 has added native support through its `<audio>`, `<video>`, and `<canvas>` tags.

In HTML4

HTML4 provides no native means to play video or audio, which forces users to rely on proprietary third-party software players. These players are typically delivered through browser plug-ins such as Adobe Flash, Microsoft Silverlight, and Apple QuickTime.

The necessity for third-party software to play web video has come under fire in the technical community. In April 2010, Steve Jobs published a letter on the Apple corporate website citing his reasons for not supporting Adobe Flash on Apple devices.⁵ Several of Jobs' main critiques apply equally to any third-party software utility:

1. Adobe Flash is *proprietary software*, meaning that Adobe will determine the future enhancement and pricing of Flash. The proprietary nature of Flash makes it ineffective for the Web, especially given the universal nature of web video. Web technologies should remain open and determined by a standards committee like the W3C, not by a single company.
2. Adobe Flash suffers from *performance and security concerns that only Adobe can remedy*. If Adobe does not act on these concerns in a timely manner, all Flash users are at risk.
3. Flash *hinders the development of third-party web applications* by inhibiting the libraries and features that developers can use. Flash is meant to be a cross-platform tool, so Adobe might not adopt enhancements that apply only to Apple devices until those enhancements are also available on all platforms. This hinders Apple's ability to provide the most innovative applications to its users.



Figure 5. Entering a telephone number on a mobile device using Safari. Telephone-only input values make it more natural to enter the correct input.

In HTML5

HTML5 includes native support in the form of `<audio>` and `<video>` tags for both audio and video elements, thus excluding the need for software installations like Adobe Flash. Because HTML5 multimedia elements are native HTML, developers can integrate them within other page elements using CSS and JavaScript. Website designs can couple visual effects like shadows, borders, and shape transformations with control effects like adjusting playback rates or synchronizing videos to create a novel web multimedia experience.

Because multimedia playback controls are no longer tied to third-party plug-ins, developers can blend these controls seamlessly with the site's design. Audio and video are no longer simply external resources on a page, but have become elements that developers can interweave into the web application's core.⁶ With no third party software, multimedia can perform on devices that conform to web standards, such as the Apple iPad and iPhone.

The HTML5 `<canvas>` tag offers an entirely new, native way to output visual data to the browser screen. The `<canvas>` tag provides a surface on which developers draw figures; create animations; and combine text, images and animations—all through JavaScript and within the browser.

In Figure 6, because both the audio and canvas elements are HTML5 elements they can be integrated; the cube pulsates and rotates in response to the audio stream. Implementations of this functionality used to require a third-party plug-in like Adobe Flash or Microsoft Silverlight. Canvas is suitable for a diversity of applications, from games and graphs to complete web applications.

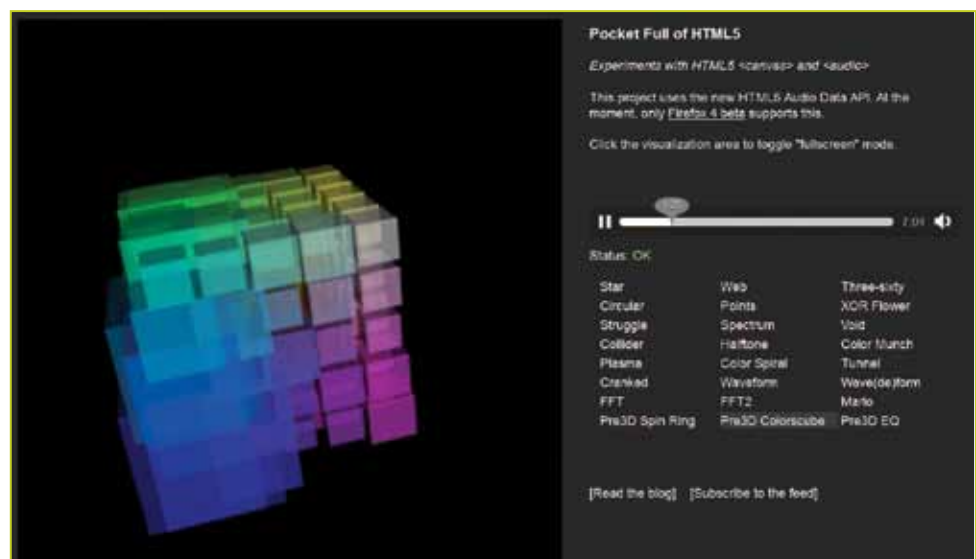


Figure 6. Freeze-frame of audiostream visualization using HTML5's `<canvas>` and `<audio>` tags. The cube pulsates in synchronization with the audio. `<canvas>` is one of several HTML5 tags that lets website designers integrate multimedia elements in new ways. (Image courtesy of J. Seidelin; www.nihilogic.dk/labs/pocket_full_of_html5.)



The Bottom Line

HTML4 requires proprietary third-party software, such as Adobe Flash, to play audio and video files. The lack of native browser support decouples audio and video elements from the rest of a site's design. HTML5's new tags fully support audio and video elements, allowing designers to tightly integrate them into a site and try bold new multi-media experiments.

New application programming interfaces

HTML5 offers new scripting APIs—drag and drop, storage, and geolocation—which allows HTML5-compliant browsers to become more like programming platforms rather than simple browsing utilities. These APIs close the functional gaps between desktop and web applications, while exploiting the unique capabilities of mobile devices.

Drag and drop

The drag and drop API enables the familiar desktop action of drag and drop to move files, attach documents, and manipulate interfaces. As such, the API provides users with a certain comfort level as they move objects within a web application in the same way that they would move them on a desktop. With HTML4-based sites, users must often traverse complicated menus to perform similar actions. This API is already appearing in popular services such as Google's Gmail, which uses it to facilitate adding email attachments.

Local and offline storage

An unfortunate byproduct of increased user mobility is the potential lack of a persistent data connection. HTML5 provides a local and offline storage API that lets web applications continue to run locally when an Internet connection is unavailable. By caching application data and automatically updating that cached data as changes occur, HTML5-compliant web browsers can redirect users automatically to local files. Once a connection is restored, web applications can resynchronize cached data with a remote server, updating application data as necessary. For the first time, users will have an uninterrupted work state for web applications—one that mimics the functionality of desktop applications.

Geolocation

Geolocation uses Global Positioning Service (GPS) coordinates to determine the user's immediate location, allowing service providers to send the user anything from local weather to the closest gas station.

Many businesses have successfully integrated geolocation into their services to improve their products' accessibility and their customers' experience. For example, Yelp users can browse the ratings of a nearby business, such as a restaurant, bar, or entertainment venue, on the basis of their mobile device's GPS coordinates.

HTML5's geolocation API permits a similar functionality within compliant web browsers, which organizations can couple with common mapping utilities like Google Maps to direct users to an office that serves their location or relevant service need.

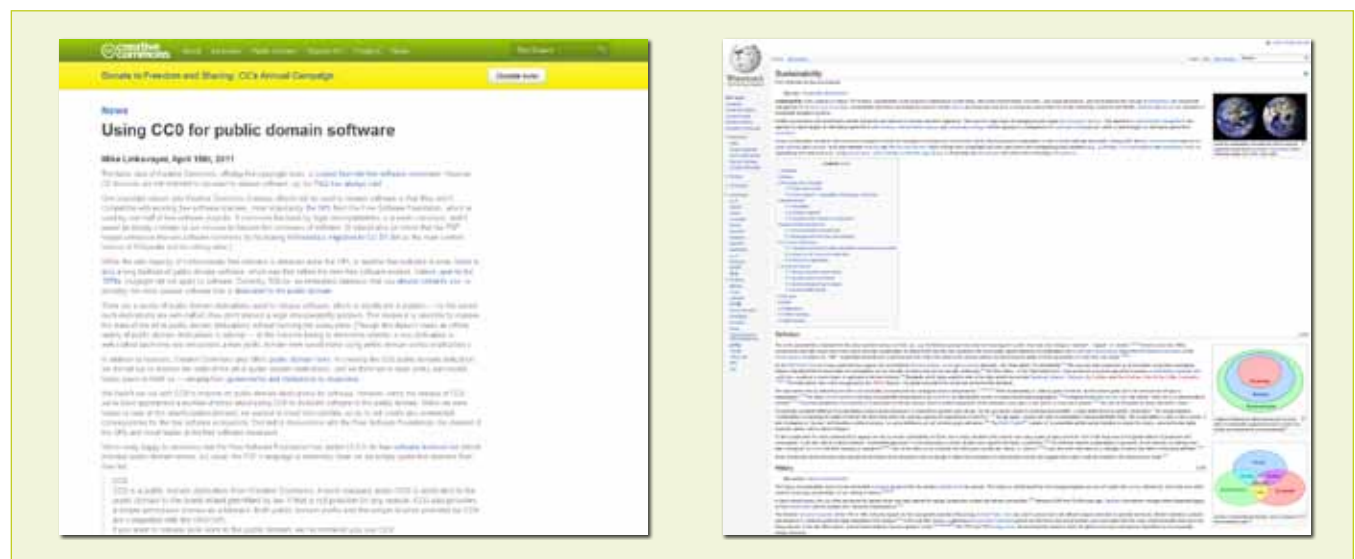


Figure 7. Structural limitations in HTML4. Both pages display news articles, but their page structures differ in a nonstandard way. Both are encapsulated in the generic `<div>` tag, and both `<div>` tags have associated ids. However, the ids are inconsistent even though both pages display article content. The first (M. Linksvayer; <https://creativecommons.org/weblog/entry/27081>) uses `<div id='post'>`; the second (<http://wikipedia.org/wiki/Sustainability>) uses `<div id='bodyContent'>`, which creates an inconsistency.

Three Cornerstones of HTML5 Adoption

1. **Use Modernizr to detect HTML5 support.** For example, if users cannot view HTML5 audio/video, the code can instruct the site to serve audio/video using Adobe Flash.
2. **Consider HTML5 features when designing web applications.** HTML5 application programming interfaces provide features that are novel to the Web. They essentially free developers from old notions of what a web application can do.
3. **Adopt HTML5 semantic markup.** HTML5 is fully backward compatible, so adopting HTML5 markup now saves on development costs later while not impacting page functionality.



The Bottom Line

By providing users with familiar behavioral and functional features like drag and drop and offline storage, HTML5's new APIs help bridge the functional gaps between desktop and HTML4 web applications. By incorporating features like geolocation, sites can reach the growing diversity of available mobile device users.

Search models for the semantic web

Arguably HTML5's most significant contribution is its reflection of the shift in the Internet's structure and purpose. Web 1.0 launched the model of webpages with static content that rarely changed. Web 2.0 marked the transition of the Web to a social and interactive space. Web 3.0 is yet another shift, this time to a more efficient way to handle massive volumes of data.

Organizations such as Google, Microsoft, and Yahoo have created entire businesses around search technology and the optimal way to parse large datasets into information. Although these businesses have proven tremendously successful to date, HTML4 creates inherent structural limitations for the webpage indexing tools that search engines use. HTML5 offers a standardized, meaningful page structure, so search engines can rapidly and correctly index pages.

In HTML4

The source of search-engine difficulty is that the same HTML4 markup tags are used to implement different page areas regardless of that area's purpose. The screenshots in Figure 7, for example, represent articles from two webpages, but the site creators used the same <div> tag to render both text blocks, even though each clearly has a different purpose.

Because <div> tags are used generically within webpages to arrange content of all types—including sidebars, titles, navigation, and menus—it can be a challenge for search engines to determine where the content within an article begins and ends. As a result, it is possible to associate words that appear in

proximity to one another but are in fact separate. For example, many navigation menus on news websites are coded using <div> tags. Because a <div> tag would similarly bind article text, a search engine cannot easily determine that there is no association between the text from the navigation menu and the text of the article on a page. Without tags that capture the different respective semantics, such distinctions are problematic.

In HTML5

HTML5 eliminates this ambiguity by introducing tags that implement semantic markup, which meaningfully dictates a webpage's structure. These structural enhancements are part of the Web 3.0 vision of Tim Berners-Lee, the Web's inventor and the current W3C director. Berners-Lee maintains that by semantically marking the structure of data found on websites, people and computers will more rapidly connect datasets and generate knowledge.

Several HTML5 tags, such as <article> make it easier for search engines to group similar content.⁷ In Figure 7, rendering article text blocks using the <article> tag within the pages eliminates any ambiguity about the limits of each article's content.

Other semantic additions to the HTML5 specification include <aside>, <figure>, <footer>, <header>, <hgroup>, <nav>, <section>, <time>, and <mark>.



The Bottom Line

HTML4 provides no standard way to display web data, so there is no standard way to interpret it. Websites that implement semantic markup through HTML5 can help indexing and interpreting tools realize greater operating efficiencies. Users benefit by getting relevant information faster. Organizations benefit by disseminating information to requesters with greater precision.

Growing pains

The HTML5 specification remains a work in progress. Although video, audio, canvas, and geolocation are operational, local and offline storage is still in development. However, web browsers remain a significant barrier to the accessibility of HTML5 features. As of August 2011, Opera is the only major browser supporting native HTML5 input, although all major browsers have plans to adopt the specification (<http://html5test.com/results.html>).

HTML5 audio and video elements have similar complications. Although the native HTML5 <video> tag exists for all major browsers, support for video file types remains browser specific,⁸ and audio support is based on file type and browser.⁹ To accommodate the variety of file types and browsers, web developers must code appropriate solutions.

Consequently, a solution that combines HTML5 audio, video, and canvas may not yet be sufficient to replace all applications of third-party multimedia plug-ins. Software like Adobe Flash and Microsoft Silverlight include features that are still under development for canvas, including hardware acceleration and 3D support. As support for canvas and HTML5 multimedia grows this should change.

Implementation steps

Organizations can take concrete steps to begin adopting HTML5 technologies now. The sidebar, “Three Cornerstones of HTML5 Adoption” outlines three foundational adoption strategies. The first is to have some way of detecting HTML5 support within a user’s web browser. Sites can use tools like Modernizr, an open-source JavaScript library, to test for a browser’s support of specific HTML5 functions. To accommodate users whose browsers lack that support, developers can code in alternative paths that maintain a site’s intended functionality, such as redirecting users to Flash-based audio or video.

Fortunately with the rapid progression of HTML5 development, it will not be long before HTML5 matures enough to provide complete cross-browser support. Organizations that implement HTML5 features now can simply remove Modernizr detection later to have a fully functional HTML5 website.

The second adoption cornerstone is to be aware of all HTML5 features when planning web applications, particularly in light of how they can enrich their customers’ user experience. Offering functionality that mimics desktop applications, such as drag and drop and local and offline storage, is an easy way to instill familiarity with and trust in web applications. By adding native HTML5 inputs that render according to input type, web applications can preemptively protect users against entering incorrect input, further solidifying that trust. Incorporating these concerns early in the software life cycle will help organizations keep usability considerations at the forefront of product development.

The last adoption cornerstone is to adopt semantic markup. This is particularly important for organizations that want more exposure through search engines and indexing utilities. For example, sites like data.gov (www.data.gov/about) aggregate machine-readable datasets from organizations across the federal government.

“Giving Data a New Face” on p. 33 describes the ease of extracting information from websites that display data semantically and how users are more likely to translate data visualizations into actionable information. Organizations that exert the effort to create and analyze data should reap the benefits of having that data discoverable by other sources.

The development of HTML5 aims to fill the gaps left by the Internet’s explosive growth over the last decade. With the expansion of Internet-driven applications, in addition to the growth of devices like smartphones and tablets, it quickly became evident that the Web’s core functionality required an expansion in kind. HTML5 presents a more usable feature set to next-generation Internet users while capitalizing on the Web’s growth. By understanding and implementing the features outlined within the HTML5 specification, organizations can better position themselves to serve their interests, engage customers, and remain on the cusp of web technology. ■

References

1. *The Size of the World Wide Web (The Internet)*; www.worldwidewebsize.com.
2. *Introduction to HTML4*, World Wide Web Consortium (W3C); www.w3.org/TR/html401/intro/intro.html.
3. R. Johansson, *HTML5 Input Types*, 456 Berea Street; www.456bereastreet.com/archive/201004/html5_input_types.
4. *Cisco Visual Networking Index: Forecast and Methodology, 2010–2015*, June 1, 2011; www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html.
5. S. Jobs, *Thoughts on Flash*, Apr. 2010; www.apple.com/hotnews/thoughts-on-flash.
6. R. Paul, *HTML5 and Web Video: Freeing Rich Media from Plug-in Prison*, May 2009; <http://arstechnica.com/open-source/news/2009/05/google-dailymotion-endorse-html5-and-standards-based-video.ars>.
7. *W3C Semantic Web Activity*; www.w3.org/2001/sw.
8. *Semantics—Dive into HTML5*; <http://diveintohtml5.org/semantics.html>.
9. M. Boas, *Native Audio in the Browser, HTML5 Doctor*, Oct. 14, 2010; <http://html5doctor.com/native-audio-in-the-browser>.

Daniel Y. Na’s bio and photo are on p. 26.



Brian C. DeRocher is a lead staff member at Noblis, where his experience includes systems administration and programming. He received a BS in mathematics and computer science from the University of Maryland. Contact him at brian.derocher@noblis.org.

Noblis ISO 9001 Certified



Providing quality, professional support for our clients has always been a top priority for Noblis. Our efforts were recently recognized by the International Organization for Standardization (ISO) when we became ISO 9001 certified. This certification demonstrates that the strength of our quality management practices permeates every aspect of our organization.

Fusing Business Process Management and the Social Web

David W. Vera

The federal government is facing challenges of vast complexity. A cohesive blend of BPM and mobile and social technologies can provide a mechanism to address those challenges.

The confluence of several governmental initiatives and priorities—from operational efficiency and cost reduction to customer service and knowledge retention—has created a need for innovative approaches to the way agencies plan and implement IT solutions. Many government agencies are using business process management (BPM) as a component in automating workflow and tasks, increasing standardization, and improving performance. BPM combined with service-oriented architectures provides the extension to business capabilities and asset reuse that organizations desire.

The emergence of social web juggernauts, such as Twitter and Facebook, has accelerated large-scale mobilizations, broadened movements, and greatly expanded access to both topical information and knowledge about major events. However, very few agencies are leveraging the power of the social web inward to support creative problem solving and to capture the tacit expert knowledge that is shared around complex processes.

The key to harnessing this power is to recognize the growing symbiotic relationship between mobile and social technologies and its multiplier effect on user adoption.

Fusion in action

Noblis is partnering with its clients to develop a strategy that builds on and merges traditional BPM and workflow concepts with social web and mobile paradigms. This strategy will empower the federal workforce to efficiently create, track, collaborate on, and manage a variety of work items throughout the items' life cycle.

In one effort, Noblis is applying a fusion strategy to help a government agency's planning to improve its business operations and customer service. Figure 1 shows the components of the envisioned solution, which would allow field technicians to initiate a complaint-intake process on a mobile device, collect the necessary data, and submit the complaint directly to the system remotely.

The system will analyze the complaint details and use predefined business rules to flag the newly submitted complaint as a high-priority item that requires the field manager's immediate review and approval. The field manager then receives a notification on his or her mobile device that a high-priority complaint requires review and action.

For example, the manager might open the complaint, review the complaint details, and realize that the regulations dictating the complaint's validity have recently undergone revision. The manager must then consult and collaborate with geographically-dispersed domain experts as a part of their decision making

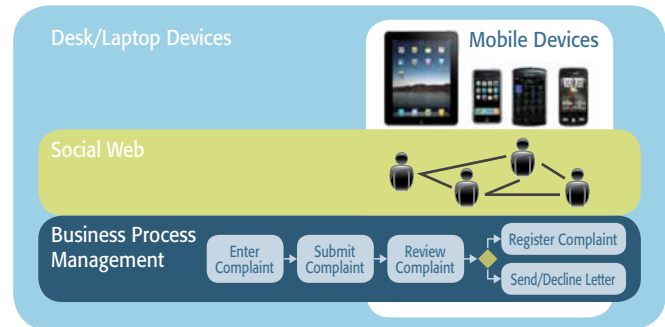


Figure 1. Fusing BPM, the social web, and mobile devices to address high-priority customer complaints. Such fusion saves time and cost in meeting customer service needs.

process for accepting the complaint as a case. The manager can share the review task with several identified regulatory experts and invite them to collaborate around the complaint work item to determine the appropriate course of action. If they accept the invitation, the regulatory experts in turn can share their insights and knowledge about the regulatory revisions.

The virtual teaming and collaboration takes place within the application around the work item and leads to a consensus on how to move forward. In this case, the manager reviews the determination through the mobile device's rich, context-sensitive interface and with a touch can keep action flowing through the process to approve and register the complaint. The customer's complaint is registered and the customer receives an email notification to a provided address stating the outcome of the complaint submission and any subsequent actions to be taken.

Expected benefits

As this one example shows, the benefits resulting from a unified strategy to fuse technologies will be significant. *Agencies* will improve the quality and consistency of business operations. Capturing collaborative details will transform work items into organizational knowledge assets. Last but not least, agencies will be better able to extend the reach of mobile government workers.

Business users will also benefit by reduced learning curves and increased productivity from the process knowledge codified in business applications that direct workflow. Knowledge sharing will also encourage creativity and innovation.

Arguably *customers* stand to benefit the most from a fusion strategy, since it will enhance the consistency, timeliness, and quality of their interactions with and services received from government agencies. Agency collaboration and service delivery will make it possible for customers to access information at any time and in any place. Finally, improved information reliability and availability is likely to result in more positive outcomes. ■

For more information, contact David Vera at david.vera@noblis.org.



With our partners Cray Supercomputers, the Commonwealth of Virginia, the Virginia Tobacco Commission, and the City of Danville—Noblis announces the

Center for Applied High Performance Computing

A transformational center accelerating the development of high performance computing applications to solve big data and advanced analytics problems. The Center will house the first next generation Cray XMT™ supercomputer in the U.S. outside of a federal lab or university.



For the best of reasons



3150 Fairview Park Drive South
Falls Church, VA 22042

noblis.org

