Web Site Mobilization Techniques

John Christopher Sandvig
Western Washington University, USA

INTRODUCTION

The introduction of the first Apple iPhone in January 2007 started a historic shift in web browsing from desktop and laptop computers to smartphones and other mobile digital devices. By 2014 two of the largest global shopping sites, Amazon.com and Target.com, reported that the majority of their traffic originated from mobile devices (Sterling, 2015). During the first quarter of 2015 34% of global e-commerce transactions were conducted via mobile devices ("State of Mobile Commerce," 2015). The number of smartphone users worldwide is forecast to grow from 2.08 billion in 2016 to 2.5 billion by 2019. Over 36% of the world's population is projected to use a smartphone by 2018 (Statista, 2016). As the number of smartphone uses grows it is becoming increasingly important that web sites provide a good user experience on mobile devices.

BACKGROUND

Mobile devices have several physical limitations that require the use of special web design techniques to produce a mobile-friendly user interface. These physical limitations include small screens, virtual keyboards, slow download speeds, and high network latency. To accommodate these physical limitations mobile-friendly web sites often display less information than desktop sites, have larger tap targets such as buttons and form fields, use smaller images, and offer fewer navigation options. Sites that are developed primarily for desktop and laptop computers often require mobile users to zoom in and scroll horizontally, providing a poor user experience. Mobile-friendly web sites are designed to render well on all devices, not just mobile devices.

MOBILIZATION TECHNIQUES

Currently there are three popular techniques for developing mobile-friendly web sites: responsive, dynamic serving, and redirect to separate URL. These techniques are referred to by a variety of names so to avoid confusion this article will adopt the nomenclature utilized in Google's Mobile Developer Guide ("Mobile SEO Overview," 2016). All three techniques detect the size of the user's screen and modify the page content and layout to accommodate the device. The techniques differ in how they detect screen size and the mechanisms by which they modify page content.

Responsive Web Design

The responsive design technique delivers the same content (HTML, CSS, JavaScript, images, etc.) to all devices and "responds" to screen size by modifying the layout. For instance, a page design that utilizes a three column layout on a desktop computer may be displayed as one long vertical column on a mobile device. Tabbed navigation on a desktop computer many be displayed as a hamburger menu on a mobile device, and large images may be reduced in size for mobile devices.

Responsive is a relatively new technique. The phrase "response web design" was originally defined by Ethan Marcotte in 2007 (Els, 2015). Marcotte states that web pages should be designed with the flexibility to respond and adapt to the capabilities of various devices. At the time Marcotte proposed this approach CSS did not have the

DOI: 10.4018/978-1-5225-2255-3.ch704

ability to detect screen size but the approach could be implemented using JavaScript. This capability was included in the final recommendations for CSS3 and adopted in 2012 by the World Wide Web Consortium (Hargreaves, 2015). It then took a few years for web browser software to implement new standards and for web site developers to employ the new capabilities in web sites.

CSS media queries have the ability to detect screen size, thus allowing developers to create different layouts for different size screens. For example the following media query located in a site's css file would allow a developer to define styles for devices with screen widths between 320px and 400px.

```
 @media screen and (min-device-width:
320px) and (max-width: 400px) {
        /* styles for small screens
go here */
}
```

The responsive technique is often implemented using any one of several open-source grid frameworks. These frameworks provide a basic structure for page layout, eliminating the need for developers to create designs from scratch. There are over two dozen grid frameworks include Bootstrap, Foundation 3, Skeleton, 1140 CSS Grid, and Less Framework 4 (Jain, 2015).

CSS frameworks allow developers to design different layouts for different size screens. For instance, Bootstrap 4.0 defines breakpoints at five different screen widths: extra small (less than 34em), small (34-47em), medium (48-61em), large (62-74em) and extra large (75em and larger) (Oliver 2015). Web designers can then create specific layouts for each of these five screen size ranges. Other styles classes, such as Bootstrap's. hidden-*-up and.hidden-*-down (where * is xs, sm, md, lg, and xl) allow developers to target specific content to different size screens. Many grid frameworks also provide JavaScript files that work in conjunction with built-in CSS classes to provide dynamic features such as dropdown navigation, breadcrumbs, and hamburger menus (Lambert, 2016).

The responsive mobilization technique can also be implemented using JavaScript. JavaScript is a programming language that is supported by all modern browsers. It has the ability to detect screen sizes and apply formatting styles. For example the following JavaScript would hide an HTML element for screens widths between 320px and 400px.

```
if(screen.width  >=  320 && screen.
width <= 400) {
    document.getElementById("nav").
style.display = "none";
}
```

While JavaScript is capable of implementing the responsive technique CSS media queries are designed specifically for page formatting and are easier to implement (Knight, 2011).

Dynamic Serving

The dynamic serving technique sends different page content and layout to different devices. The web server examines information which is contained in the web request and sends content and layout appropriate to the capabilities of the device. The dynamic serving technique has the advantage that it can send assets which are optimized for the device. For instance, if the server detects a mobile device it may send smaller and fewer images, thus reducing data usage and page load time.

To identify the requesting device the dynamic serving technique examines metadata that is included in the headers of each web request. Metadata includes information about the request, such as character encoding, language, cookies, caching, and the user-agent. The user-agent attribute includes information about the user's