



ESE Kongress 2013

Vortragsskript:

Apps plattformunabhängig entwickeln Möglichkeiten und Grenzen von HTML5 und JavaScript

Frank Listing, MicroConsult GmbH

Applikationen unabhängig vom Betriebssystem entwickeln, ohne dass eine spezielle Laufzeitumgebung installiert ist – das wäre doch super. Jeder Produktmanager träumt davon. Warum ersetzen wir nicht das Betriebssystem durch den Browser? Den gibt es überall. Jeder Browser versteht HTML, und seit vielen Jahren kann jeder Browser JavaScript. Damit ist doch alles vorhanden. Wir beschreiben die Benutzeroberfläche mit HTML und programmieren mit JavaScript. – Alles wird gut: schnellere Entwicklungszeiten, billige Entwickler (man muss ja keine exotischen Programmiersprachen beherrschen) und keine Sonderlösungen für eine spezielle Plattform.

Das klingt alles ganz toll. Aber wie immer steckt der Teufel im Detail. Deshalb wird in diesem Artikel der Mythos HTML5 etwas näher betrachtet und die Möglichkeiten und Grenzen der Applikationsentwicklung speziell für mobile Geräte werden aufgezeigt.

HTML5

Bei dem Begriff HTML5 geht die Verwirrung schon los. An sich ist es nur die Auszeichnungssprache für die Erstellung von Webseiten. Allerdings wird landläufig unter dem Begriff „HTML5“ eine Sammlung von Technologien verstanden, die zwar im Zusammenhang mit HTML(5) verwendet werden, aber nicht HTML sind, wie z.B. Zugriff auf GPS und lokale Datenspeicherung.

Schon die Entstehung von HTML5 ist recht interessant. Eigentlich sollte es gar kein HTML5 geben. Das W3C (WWW-Konsortium) favorisierte XHTML 2.0, das sich vor allem durch eine saubere Syntax auszeichnet. Durch die Verwendung von XML als Basis ergibt sich aber ein entscheidender Nachteil: Ein kleiner Fehler verhindert das Laden der gesamten Webseite. Das ist im praktischen Einsatz nicht effizient.

Auch sonst hat das W3C wohl etwas abseits der Realität gearbeitet. So haben die führenden Browser-Hersteller die Initiative ergriffen und angefangen HTML5 zu entwickeln. Das Ganze passiert im Rahmen der WHATWG (Web Hypertext Application Technology Working Group).

Kernpunkte von HTML5

- Dynamische Erzeugung von Grafiken (2D, 3D, Vektorgrafik) mit JavaScript
- Einführung semantischer Elemente, um komplexe Webseiten zu strukturieren
- Einbinden und Abspielen von Audio- und Video-Dateien
- Zugriff auf das Dateisystem, um auch auf lokale Dateien zugreifen zu können
- Möglichkeiten, mit denen Applikationen über mehrere Browserfenster hinweg kommunizieren können

Langfristig sollen Web-Applikationen die Leistungsfähigkeit von nativen Applikationen erreichen.

Im Jahr 2007 begrub das W3C XHTML und begann, HTML5 zu standardisieren. Damit gibt es jetzt zwei HTML5-Standards, den vom W3C und den der WHATWG. Der Standard des W3C ist im Wesentlichen eine Untermenge des WHATWG-Standards.

Zu dieser Problematik kommen noch die Interessen der einzelnen Browserhersteller. Sie implementieren unterschiedliche Teile der Standards, und die Qualität der unterschiedlichen Implementierungen schwankt stark.

Bei allen Schwierigkeiten bietet HTML5 jedoch viele interessante neue Möglichkeiten. Allerdings sind diese nur im Zusammenhang mit zwei anderen Technologien wirklich nutzbar: CSS (Cascading Style Sheets) für die Formatierung der Seite und JavaScript für die Programmierung.

CSS

CSS (Cascading Style Sheets) definieren das Aussehen der Elemente einer Webseite. Sie ermöglichen die Trennung von Inhalt und Design. Die Flexibilität bei der Erstellung einer Webseite wird erhöht. Stylesheets beeinflussen die Präsentation eines HTML-Dokuments, ohne das Dokument selbst zu verändern. CSS sind ein W3C-Standard.

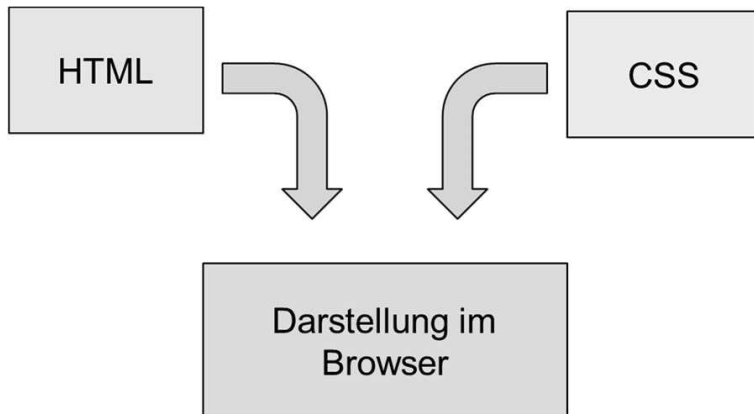


Bild 1: Cascading Style Sheets

In einem Stylesheet werden Regeln definiert, die sich auf ein oder mehrere Elemente der HTML-Datei auswirken. Das sind z.B. Angaben zur Schrift (Art, Größe, ...), Abstände, Rahmen usw. Im neuesten Standard (CSS3) sind sogar Transformationen (Rotieren, Skalieren, Kippen, Versetzen) und Animationen enthalten.

JavaScript

JavaScript ist eine plattformübergreifende Skriptsprache. Sie ist nicht für die Verwendung als eigenständige Programmiersprache gedacht, sondern als eingebettete Sprache z.B. für Webbrowser und ist heutzutage in jedem Browser enthalten.

JavaScript wurde 1996 von Netscape mit dem Navigator 2.0 veröffentlicht. JavaScript hat mit Java nicht viel zu tun. Der Name wurde gewählt, weil damals angenommen wurde, dass Java die Zukunft gehört. Der Markenname JavaScript ist Eigentum von Oracle (seit der Übernahme von Sun Microsystems). Deshalb erfolgt die weltweite Standardisierung von JavaScript unter dem Namen ECMAScript (ECMA-262).

Die Sprache selbst ist einfach aber objektorientiert aufgebaut und lehnt sich in der Syntax an die Sprachen der C-Linie an. Der Kern von JavaScript beinhaltet die grundlegenden Syntaxelemente wie Schleifen, Bedingungen etc. und einen grundlegenden Satz von Objekten (z.B. Array, Date). Die Applikationen, in welche JavaScript eingebettet wird, können beliebige zusätzliche Objekte bereitstellen (z.B. document im Webbrowser).

JavaScript gehört zu den schwach getypten Sprachen, d.h. viele Fehler werden erst bei der Ausführung des Skriptes gefunden. Mit JavaScript lässt sich auf die Elemente des Dokumentenbaums des HTML-Dokuments zugreifen. Elemente können entfernt, hinzugefügt oder verändert werden. Damit kann eine Webseite dynamisch angepasst werden. Mit Hilfe von AJAX (Asynchronous JavaScript and XML) ist es möglich, Daten nachzuladen, ohne dass die komplette Webseite neu geladen wird.

Der Zugriff auf die schönen neuen Features von HTML5, wie GPS, Grafik usw., erfolgt ausschließlich über JavaScript.

Web-Applikationen

Mit den drei Mitteln HTML(5), CSS und JavaScript können schon recht anspruchsvolle Web-Applikationen erstellt werden.

- Mit HTML und CSS ist es möglich, anspruchsvolle Oberflächen zu erstellen.
- JavaScript ist die Basis für die Dynamik der Applikation, also die Reaktion auf Benutzereingaben, Überprüfung und Aufbereitung von Daten.

Applikationen, die im Browser laufen, haben keinen vollen Rechnerzugriff.

Applikationen

Wie kommt man von der Web-Applikation zu einer „richtigen Applikation“? Im Prinzip muss nur der Browser unsichtbar gemacht werden.

Vorgehensweise:

1. Es wird eine native Applikation erstellt, deren Hauptfenster aus einer Browserkomponente besteht. Damit ist der Browser als solcher nicht mehr erkennbar.
2. Die Webseiten inklusive Stylesheets und JavaScript-Dateien werden lokal oder als Ressource der Applikation gespeichert.
3. Beim Start der Applikation wird die erste HTML-Seite geladen.

Allerdings ist die Applikation nicht mehr plattformunabhängig. Die äußere „Hülle“ muss einmal pro Plattform erstellt werden. Dieser Aufwand hält sich aber in Grenzen und kann bei weiteren Applikationen wiederverwendet werden.

Mobile Apps

Das Prinzip der eigenständigen Web-Applikationen lässt sich auch auf Apps für Smartphones übertragen. Damit sind Apps, die Benutzereingaben verarbeiten (z.B. Taschenrechner, Einheitenkonverter), schnell und kostengünstig erstellt.

Durch die Verwendung von AJAX kann weiterhin auf Daten im Inter-/ Intranet zugegriffen werden. Das vervielfältigt die Möglichkeiten enorm. Es ermöglicht die Programmierung von Auskunftssystemen oder auch die Visualisierung von Betriebszuständen von Maschinen mit Netzwerkanschluss.

Durch die Nutzung von speziellen Frameworks, wie z.B. PhoneGap oder Sencha Touch, können Applikationen auch auf Features zugreifen, die ein normaler Webbrowser nicht bietet (z.B. Kontaktdaten, Kamera). Diese Frameworks bieten zusätzliche JavaScript-Schnittstellen an, die in den nativen Wrapper mit eingefügt werden. Nachdem der plattformunabhängige Code erstellt wurde, wird vom Framework der plattformabhängige Code erstellt und kompiliert. Es muss lediglich die Entwicklungsumgebung des Zielsystems installiert sein. PhoneGap bietet sogar einen Online-Buildservice an (gegen Bezahlung). Dort werden aus dem plattformunabhängigen Code native Apps für verschiedene Zielsysteme generiert, ohne dass eine Entwicklungsumgebung installiert werden muss.

Fazit

Die Entwicklung mit HTML5 und JavaScript wird die native Programmierung nicht ersetzen, sondern ergänzen. Vor allem Applikationen, die auf keine lokalen Daten zugreifen, sind recht einfach zu erstellen. Als einziges Problem taucht die nicht einheitliche Umsetzung des HTML-Standards durch die unterschiedlichen Browser auf. Das kommt je nach Oberflächengestaltung mal mehr oder weniger zum Tragen.

Quellen

<http://phonegap.com/>

Florian Franke, Johannes Ippen „Apps mit HTML5 und CSS3: für iPad, iPhone und Android“, Galileo Computing

Matt Gifford „PhoneGap Mobile Application Development Cookbook“, Packt Publishing

Autor

Dipl.-Ing. Frank Listing ist seit 2002 Trainer und Projektcoach bei der MicroConsult GmbH mit dem Schwerpunkt Microsoft-Plattformen, objektorientierte Programmierung und Testen von embedded Systemen und u.a. fachlich für das Thema .NET verantwortlich. Sein Wissen gibt er immer wieder auch in Publikationen und Fachvorträgen weiter.