MASTER THESIS

zur Erlangung des akademischen Grades
„Master of Science in Engineering“
im Studiengang Multimedia und Softwareentwicklung

HTML5 Canvas vs. Adobe Flash

Ausgeführt von: Peter Kerschner, BSc.
Personenkennzeichen: 1210299029

1. BegutachterIn: Dipl.-Ing. (FH) Arthur Michael Zaczek
2. BegutachterIn: Dipl.-Ing. Mag. Dr. Michael Tesar

Wien, 9. März 2014

FACHHOCHSCHULE
TECHNIKUM WIEN

Eidesstattliche Erklärung

„Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbständig angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher weder in gleicher noch in ähnlicher Form einer anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht. Ich versichere, dass die abgegebene Version jener im Uploadtool entspricht."

Ort, Datum

Unterschrift

Kurzfassung

Die Neuerungen von HTML5 bieten Entwicklern neue Möglichkeiten. Mit Hilfe des Canvas-Elements und der Verwendung der Skriptsprache JavaScript kann der Browser für Multimedia-Anwendungen und Spiele genutzt und damit die weitere Internet-Entwicklung unabhängig von Adobe Flash gemacht werden. In Zukunft könnte das bedeuten, dass Flash-Anwendungen obsolet werden.

Diese Arbeit befasst sich mit der Gegenüberstellung und dem Vergleich der beiden Kerntechnologien HTML5 und Adobe Flash. Dabei soll aufgezeigt werden, wie sich der Einsatz von HTML5 und Adobe Flash aktuell gestaltet und welche der beiden Technologien ein größeres Zukunftspotenzial besitzt. Anhand von praktischen Umsetzungen sollen die Vorteile und Nachteile beider Konkurrenten aufgezeigt und ein grober Leitfaden gegeben werden, für welche Projekte welche Technologie bevorzugt eingesetzt werden kann.

Nachdem zu Beginn die technischen Grundlagen vermittelt werden, folgt ein Vergleich zwischen den beiden Technologien HTML5 und Adobe Flash. Im Anschluss werden Analysen anhand von praktischen Umsetzungen im Bereich Multimedia- und Spiele-Entwicklung durchgeführt. Abschließend wird das Ergebnis diskutiert und ein Blick in die Zukunft geworfen.

Schlagwörter: Web-Entwicklung, Web Standards, HTML5, Adobe Flash, CSS, JavaScript, ActionScript

Abstract

The new features of HTML5 offer new possibilities for developers. The new canvas element in combination with JavaScript can be used to develop multimedia applications and games for the browser. The result is the progress of the internet independent from Adobe Flash, which possibly means that Adobe Flash can be replaced by HTML5.

This paper compares the core technologies of HTML5 and Adobe Flash, pointing out how HTML5 and Adobe Flash can be used at present and which of the two technologies has a greater potential for the future. On the basis of practical implementations, the advantages and disadvantages of both competitors are shown and rough guidelines are be given for the choice of which technology can be preferably used for which project.

Analyses based on practical implementations in the field of multimedia and games include a discussion of the results and future perspectives of the two technologies.

Keywords: Web-Entwicklung, Web Standards, HTML5, Adobe Flash, CSS, JavaScript, ActionScript

Danksagung

———DANKSAGUNG———-

Inhaltsverzeichnis

1. Einleitung

Wahrscheinlich gibt es keine andere Erfindung, abgesehen von der Telekommunikation und des Fernsehens, die die Gesellschaft der Industrie- und Schwellenländer entscheidender prägt als das Internet. Das World Wide Web, der bekannteste Dienst des Internets, erlebt einen stetigen Anstieg an Nutzerzahlen. Immer mehr und vor allem jüngere Nutzergruppen lassen sich durch soziale Netzwerke in ihren Bann ziehen und gerade diese Netzwerke haben dabei Einfluss auf die Art der Kommunikation von Generationen.

Für die Verbraucher beschäftigt sich ein kompletter Industriezweig damit, ihnen das Leben zu vereinfachen, in dem im Internet sämtliche Informationen zur Verfügung gestellt und Dienstleitungen wie diverse Einkaufsmöglichkeiten verfügbar gemacht werden. Dabei zeigt die Tendenz, dass Nutzer ihre Daten nicht mehr auf physikalischen Datenträgern speichern und transportieren, sondern diese in der "Cloud" ablegen und somit unter der Vorraussetzung eines Internetzugangs der Zugriff von überall ermöglicht wird.

Ohne der Weiterentwicklung des Internets würde es Begriffe wie „Soziale Netzwerke", „Videokonferenz", „Onlinebanking" oder „Onlineversandhaus" nicht geben - Begriffe, die vor allem in den vergangenen Jahren stetig an Bedeutung gewonnen haben. Wohin die Entwicklung des Internets dabei führen wird, lässt sich kaum vorhersagen. Fest steht, dass das Internet ein wesentlicher Teil im Leben der aktuellen Generation einnimmt und für zukünftige Generationen einnehmen wird.

Seit der Gründung des World Wide Web Consortiums (W3C) am 1. Dezember 1994 wird bemüht an der Umsetzung eines einheitlichen Standards und dem Festlegen der dazu notwendigen Rahmenbedingungen gearbeitet. Seit Jahren wird weltweit solch ein neuer Standard von Webentwicklern herbeigewünscht, um unnötigen Programmieraufwand aufgrund unterschiedlicher Browservarianten und -versionen zu beseitigen.

Die vollständige Erarbeitung und Etablierung dieses Standards wird allerdings noch einige Jahre in Anspruch nehmen. Dabei liegt die Problematik darin, dass der neue Standard für die Zukunft gewappnet und die Bereitstellung älterer Webseiten kompromisslos möglich sein muss. Dieser doppelte Anspruch verzögert die Fertigstellung ungemein.

1.1. Problemstellung

Die Entwickler von Flash (früher Macromedia, jetzt Adobe) haben schon früh das Potential von ihrem Flash Player erkannt, um plattformübergreifend Video- und Audioinhalte im Web bereitzustellen. Ein großer Vorteil war, dass Inhalte für Flash lediglich einmal kodiert werden mussten. Tausende von Seiten die sich für Flash als Streamingplattform von Multimediainhalten entschieden haben, bestätigten, bis heute, das Potential.

Mit dem Erscheinen von Apples iPhone und iPod touch im Jahr 2007 und der folgenden Entscheidung, Flash auf diesen Geräten nicht zu unterstützen, mussten Webseitenbetreiber darauf reagieren. Viele boten Video/Audio-Streams an, die direkt im mobilen Safari Browser wiedergegeben werden konnten. Durch die Verwendung der H.264 Kodierung konnten die Inhalte auch über den Flash-Player (sofern vom verwendeten Gerät unterstützt) abgespielt werden. Dadurch mussten Inhalte weiterhin nur einmal kodiert werden, um mit möglichst vielen Plattformen kompatibel zu sein.

Die Entwickler der HTML5 Spezifikation (und u.a. Apple) sind der Meinung, dass Browser Audio und Video nativ unterstützen sollten ohne dabei auf etwaige Plugins zurückgreifen zu müssen. Dies hat den Vorteil, dass es keine herstellerspezifischen Einschränkungen (wie es bei Flash der Fall ist) für den Entwickler gibt, was er mit den Inhalten anstellt, nachdem sie in die Website eingebettet wurden. Mittels CSS und JavaScript ist es jederzeit direkt möglich dargestellte HTML Elemente zu manipulieren.

Um ein Bild für eine Webanwendung zu erstellen wird üblicherweise eine Grafiksoftware genutzt und anschließend in die Anwendung eingebettet. Für Animationen wird vorwiegend Flash verwendet. Mit dem in HTML5 verfügbaren Canvas Element können Entwickler Bilder und Animationen direkt im Browser mittels JavaScript generieren. Mit dem Canvas Element ist es möglich, einfache bis komplexe Formen, Graphen und auch Diagramme zu erstellen, ohne auf diverse Bibliotheken, Flash oder ein anderes Plugin zurückgreifen zu müssen.

Mit der Veröffentlichung und der Implementierung von HTML5 Features in die neuesten Brower eröffneten sich neue Möglichkeiten um Animationen, Spiele und Entertainment-Applikationen für das Web zu erstellen. Aufgrund vieler auftretender Fragen ist sich kein Entwickler wirklich klar, ob HTML5 den Platzhirschen Flash verdrängen kann:

- Wo liegen die Vorteile und Nachteile beider Technologien?
- Ist HTML5 die Zukunft und löst Flash ab?
- Welche Technologie ist einfacher zu erlernen?
- Was kann HTML5 was Flash nicht kann (und visa-versa)?
- Lohnt es sich noch Flash zu erlernen?
- Gibt es Entwicklungsumgebungen, die das Programmieren erleichtern?

1.2. Motivation

Das Internet stellt den weltweit größten Netzverbund und eine Zusammenfassung von verschiedensten Dienstleistungen dar. Die dabei am häufigsten genutzte Dienstleistung - das World Wide Web - hat einen sehr großen Anteil an der in den 90er Jahren enstandenen und stetig wachsenden Beliebtheit des Internets. Seit dem hat sich das Internet vom reinen Wissenschaftsnetz zu einem kommerziell genutzten Netz, mit einer Vielzahl an Diensten und multimedialen Anwendungen, entwickelt.

Für viele Nutzer ist das Internet fast nicht mehr aus ihrem Leben wegzudenken. Zu verlockend sind die Möglichkeiten der flexibleren Gestaltung des Privatlebens, Zeitersparnisse, Kontaktpflege mit Familie und Bekannten, Arbereitserleichterung und vielem mehr. Die Möglichkeiten, die das Internet bietet, sind fast grenzenlos.

Für eine Menge Menschen bilden soziale Plattformen, wie Facebook oder Google+, die Grundlage ihrer sozialen Interaktion und dementsprechend hoch sind die Ansprüche. Generell steigen mit der fortgeschrittenen Entwicklung des Internets und dessen Technologien die Erwartungen und Ansprüche der Nutzer.

Aus diesem Grund muss schon vor dem Beginn eines Projektes sehr gut abgewogen werden, wie es umgesetzt werden soll. Eine entscheidende Rolle spielt dabei die zu nutzende Technologie aus der Sicht der möglichen Nutzer und deren Systemvorraussetzungen (Plattform, CPU- Leistung, Browser-Hersteller und -version). Andererseits stellen auch Entwicklungs- und wartungsaufwand eine nicht zu unterschätzende Größe dar.

1.3. Zielsetzung

Das Ziel dieser Arbeit ist es, zwei Technologien zu vergleichen, die in einem direkten Konkurrenzkampf stehen.

Adobe's Flash bietet einem Entwickler die Möglichkeit, mit einmaligen Entwicklungsaufwand viele Nutzer zu ereichen. Einzige Vorraussetzung ist ein installiertes Flash-Player-Plugin auf dem Computer des Seitenaufrufers. Allerdings wird genau dieses Manko bei der Entwicklergemeinde als größtes Hindernis angesehen. Wieso dem Nutzer die Installation eines Plugins aufzwingen, wenn ähnliche Funktionalitäten auch ohne angeboten werden können?

Hingegen verlangt die Verwendung bestehender Web-Standards dem Entwickler die Kenntnisse mehrerer Programmiersprachen und deren Verknüpfung ab, um annähernd vergleichbare Ergebnisse zu erzielen. Dieser Umstand und die Abhängigkeit von verschiedensten Browsern und -versionen macht es zu einer komplexen Aufgabe ein Produkt zu entwickeln, dass in allen Browsern dieselbe Funktionalität zur Verfügung stellt. HTML5 steht als Weiterentwicklung der bestehenden HTML 4.01 Spezifikation in den Startlöchern um das Web zukunftssicher zu machen. Es soll besagten Aufwand minimieren und gleichzeitig Möglichkeiten bieten, die Flash seit geraumer Zeit gewährleistet.

Deshalb gilt es in den Vergleichen herauszufinden, ob die kommenden Webstandards HTML5 und CSS3, im speziellen das Canvas Element, in Zukunft das proprietäre Webformat Flash

samt ActionScript ablösen können. Mittels der Evaluierung der Ergebnisse sollen Richtlinien erstellt werden, anhand derer Web-Entwickler leicht ermitteln können, in welchen Fällen die eine Technologie der anderen vorzuziehen ist und aus welchen Gründen.

1.4. Aufbau der Arbeit

Der Aufbau dieser Arbeit kann grob in vier Bereiche unterteilt werden. Zu Beginn bietet der Theorieteil einen Überblick und eine Einführung in die verwendeten Technologien, die für das Verständnis der Inhalte dieser Arbeit notwendig sind. Anschließend folgt ein theoretischer Vergleich beider Technologien anhand verschiedener Kriterien. Im dritten Teil folgt die praktische Analyse der jeweiligen Technologien, die Beschreibungen der umgesetzten Prototypen und deren Implementierung. Abschließend folgt eine Diskussion und Zusammenfassung der erhaltenen Ergebnisse sowie ein Ausblick auf die mögliche Zukunft des Webs und dessen Technologien.

2. Begriffserklärungen

Bevor in den nächsten Kapitel die zwei Technologien Flash und HTML5 vorgestellt und im Anschluss anhand einer prototypischen Anwendung verglichen werden, wird in diesem Kapitel zunächst die Geschichte des Internets vorgestellt. Beide Technologien nehmen einen großen Stellenwert in den Diskussionen um dieses Medium ein, wodurch diese Informationen grundlegend erforderlich sind. Im Anschluss werden die Arbeitsgruppen die sich mit HTML5 beschäftigen und Begriffe der Thematik Web 2.0 erläutert.

2.1. Die Geschichte des Internets

Der Name Internet kommt von dem englischen Begriff "Interconnected set of networks", was soviel bedeutet wie „miteinander verbundene Netzwerke". Das Internet ist heute ein weltumspannendes Netz von vielen einzelnen Computernetzwerken. Durch die Erfindung des elektrischen Stroms im Jahre 1730 war der Grundstein für die elektrische Telegrafie gelegt, welche Samuel Thomas von Soemmerring im Jahr 1809 erfand. Die Erfindung der elektromagnetischen Induktion im Jahr 1832 durch Michael Faradey führte zu den ersten Versuchen mit einem elektromagnetischen Telegrafen, wodurch 1833 die erste telegrafische Nachrichtenübertragung gelang. Ab diesem Zeitpunkt befassten sich mehrere Personen mit der Erforschung und Entwischlung von Apparaturen zur Übertragung von akustischen Signalen. Schlussendlich war es Alexander Graham Bell - im Jahr 1876 -, der die Fertigkeiten zusammen brachte, das Telefon über eine Versuchsapparatur hinaus als Gesamtsystem zur Marktreife zu bringen. Die Datenübertragung mittels Impulsen, die durch Kabel übertragen werden, stellt bis heute das wichtigste Transportmittel von Informationen dar. Aufgeschreckt durch *Sputnik*, dem ersten von der UdSSR ins All beförderten geostationären Satellit, gründete das Verteidigungsministerium der USA die"Advanced Research Projects Agency". Die ARPA hatte die Aufgabe neue Technologien im Bereich der Datenübertragung und Kommunikation zu entwickeln. Das führte Ende der 1950er Jahre, durch J. C. R. Licklider und seinem Forschungsteam, zur Entwicklung des ersten "Time-Sharing-Systems" der Welt. Dabei war ein Zentralrechner mit sternförmig angeschlossenen Terminals verbunden, wodurch mehrere Nutzer gleichzeitig dessen Rechenleistung nutzen konnten. Damit wurde es ermöglicht große geografische Distanzen mit Terminals eines Herstellers zu überwinden, allerdings war die Anzahl der Anschlüsse begrenzt. Für militärische Zwecke eignete sich dieser Netzaufbau jedoch nicht, weil eine Störung des Zentralrechner den Ausfall des gesamten Netzes bedeutete. Ein weiterführender Ansatz war das dezentrale Netzwerk, mit dem sich Paul Baran beschäftigte. Hierbei wurden mehrere Zentralrechner, die jeweils über die sternförmig angeschlossenen Terminals verfügten, miteinander verbunden. Fiel ein Zentralrechner aus, war das Netzwerk im Ganzen war geschwächt, aber immer noch in Teilen nutzbar. Um bei einem Netzwerkschaden so wenige Terminals wie möglich zu verlieren, entwickelte Pauk Barann das "Distributed Network" bei dem der Zentralrechner überflüssig wurde. Jedes Terminal, das mit dem Netzwerk verbunden war, hat alle Funktionen an Board, die für die Kommunikation im Netzwerk notwendig sind und galt somit als Computer. Ein weiterer Vorteil mehrerer Verbindungen zu einem möglichen Zielcomputer führte zur Entwicklung des "Packet-Switching", wobei eine Datei nicht mehr als Ganzes übertragen, sondern in viele Datenpakete

gleicher Größe zerteilt und einzeln transportiert wird. Staus werden somit vermieden, denn pro Paket kann die Route neu berechnet werden und muss bei einer fehlerhaften Übertragung nicht noch einmal als Ganzes übertragen werden. Diese intelligente Übertragung beschleunigte und entlastete das Netzwerk wesentlich. 1965 wurde diese Art der Netzkommunikation zum ersten Mal eingesetzt. 1966 begann die Planung zur Vernetzung aller über das Land verteilten Computerzentren der ARPA nach Paul Barans "Distributed Network". Gegen Ende 1969 wurden nacheinander die großen Computerzentren verbunden. Die angestrebte Unabhängigkeit von Herstellern und Betriebssystem führte zur Entwicklung des "Interface Message Processor", kurz IMP, durch Wesley Clark. Der IMP ist ein Minicomputer mit einem einheitlichen Netzwerkprogramm, der an jeden Großrechner angeschlossen wurde unf dür die Datenübermittlung zuständig war. Durch diese Verbindung der großen Computerzentren der ARPA enstand das ARPANET und zum ersten Mal waren Rechner verschiedenster Art miteinander verbunden. Die ersten Anwendungen im ARPANET waren "Telnet" und "FTP" bevor Ray Tomlinson 1972 eine Software zum Verschicken und Empfangen elektronischer Post veröffentlichte. Der E-Mail-Dienst war geboren und führte zu einem sprunghaften Anstieg der Nutzerzahlen. Weitere Verbreitung erfuhr das ARPANET durch Vorführung auf der International Conference on Computer Communications und Weitergabe des Wissens unter anderem an die NASA, die Air Force und Universitäten. Darauf folgte das ALOHANET, das Forschungsstationen auf Hawaii vernetzte und sich aufgrund störungsanfälliger Telefonleitungen zum PRNET (Packet Radio Network) weiterentwickelte, in dem Daten per Radiowellen übertragen wurden. Als weitere Kommunikationsmöglichkeit wurde 1973 das SATNET (Datelliten Network) entwickelt. Die Nachteile der verschiedenen Datenübertragungsarten führten schließlich 1974 Vinton Cerf und Bob Kahn zur Entwicklung eines umfassenden Netzwerkprotokolls, dass sich auf ein einheitliches Datenformat und eine einheitliche Verbindungsmethode beschränkte, dem Transmission Control Protocol / Internet Protocol, kurz TCP/IP. 1983 wurde das TCP/IP zum Standard erklärt und kommt noch heute in der Form zum Einsatz. Im selben Jahr entstand das MILNET (Military Network) in das der komplette militärische Bereich des ARPANET verlagert wurde. Das ARPANET diente ab dem Zeitpunkt nur noch rein zivilen Gruppen. Dieser Schritt war notwendig geworden, weil sich immer mehr internationale Netzwerke anschlossen und somit die Unabhängigkeit des Militärs in Gefahr geriet. Bis 1985 wurden die bestehenden Netzwerke vorrangig von kleineren und größeren Nutzergruppen zur Kommunikation und zum Austausch ihrer akademischen Forschungsthemen genutzt. 1985 gab es einen entscheidende Änderung. Die Netzwerkzugänge des 1984 gegründeten JANET (Joint Academic Network) und des NSFNET (National Science Foundation Network) wurden für alle Benutzer – egal aus welchem Forschungsbereich – geöffnet. Dadurch erlangte das NSFNET immer mehr an Beliebtheit und übernahm schließlich 1990 die Funktion des ARPANET. In diesem Schritt wurde das ARPANET eingestellt. Im selben Jahr schlossen sich die Netzwerke von Canada, Dänemark, Finnland, Frankreich, Island, Norwegen und Schweden an das NSFNET an. 1991 folgten dann Deutschland, Japan, Niederlande und das Vereinigte Königreich. Der Dienst der in der heutigen Zeit von vielen Menschen mit dem Internet gleichgesetzt wird ist das World Wide Web. Es ist 1990 aus einer Weiterentwicklung eines etwas älteren Projekts von Tim Berners-Lee – er war in den 80er und 90er Jahren Informatiker am Hochenergieforschungszentrum CERN – hervorgegangen und führte zu einer rasant ansteigenden Beliebtheit des Internet.

2.2. Browser

Ein Browser ist ein spezielles Programm, das zur Darstellung von Inhalten aus dem Internet (World Wide Web) genutzt werden kann. Er übersetzt die im Quellcode verwendeten Befehle in das

entsprechende Aussehen. Browser sind die Benutzeroberfläche für Webanwendungen, aber auch andere Daten und Dokumente können betrachtet werden. Es gibt mehrere Hersteller, die die Konventionen des W3C unterschiedlich umsetzen. Diese auseinander gehenden Herangehensweisen führen dazu, dass Webseiten in verschiedenen Browsern unterschiedlich dargestellt werden. In unterschiedlichen Versionen eines Browsers können Konventionen unterschiedlich umgesetzt worden sein, was die versionsübergreifende Entwicklung für Programmierer zu einer enormen Herausforderung macht. Zu den Merkmalen eines guten Browsers zählen Schnelligkeit, Leistungsbedarf auf dem Rechner des Anwenders und die Umsetzung der neuesten Innovationen im World Wide Web.

2.3. W3C

Das W3C ist ein internationales Konsortiom, in dem Mitgliedsorganisationen, ein fest angestelltes Team und die Öffentlichkeit gemeinsam daran arbeiten, Web-Standards und Richtlinien zu entwickeln – daher der Name World Wide Web Consortium, kurz W3C. Gegründet wurde es am 1. Oktober 1994 von Tim Berners-Lee, dem Erfinder des World Wide Web. Die Entstehung des W3C ist eng mit der Entstehung des World Wide Web verbunden. Die große Gefahr von Inkonsistenten in der Nutzung vorliegender Technologien könnte zu unwirksamen Verknüpfungen führen und das galt es möglichst zu verhindern. Die Grundherangehensweise an neue Technologien oder Teilaspekte besteht darin, den kleinsten gemeinsamen Nenner zu finden und diesen zu einer Spezifikation zu verarbeiten, so dass diese von allen Mitgliedsorganisationen unterstützt wird.

2.4. WHATWG

Die Web Hypertext Application Working Group, kurz WHATWG, ist eine Arbeitsgruppe, die von mehreren Unternehmen, darunter unter anderem Mozilla Foundation, Opera Software ASA und Apple Inc., betrieben wird und deren Ziel die Entwicklung neuer Technologien zur einfacheren Erstellung von Internetanwendungen ist. Die WHATWG wurde 2004 gegründet, weil aus Sicht der Browser-Hersteller das W3C die Entwicklung von HTML – zu Gunsten von XHTML – vernachlässigte und auf die Bedürfnisse von Programmierern zu wenig einging. Als Unterschied zu den Bestrebungen der W3C muss festgehalten werden, dass die WHATWG keinen festen Standard als Zielsetzung verfolgt. Vielmehr soll HTML weiterentwickelt werden, ohne aber auf einen definierten Stand hin zu arbeiten. Anders das W3C, dessen aktuelle Bestrebungen ganz klar auf HTML5 ausgelegt sind.

2.5. Plattformunabhängigkeit

Plattformunabhängigkeit bedeutet bei Software, dass sie ohne extra Aufwand und ohne Hilfssoftware auf unterschiedlichen Systemen ausgeführt werden kann. Genauer betrachtet ist darunter die Eigenschaft eines Programms zu verstehen, auf unterschiedlichen Computersystemen mit Unterschieden in Architektur, Prozessor, Compiler, Betriebssystem und weiteren Dienstprogrammen, die zur Übersetzung oder Ausführung notwendig sind, lauffähig zu sein.

2.6. Rich Internet Application

Der Begriff "Rich Internet Application" kurz RIA (reichhaltige Internetanwendung) wurde von Macromedia im Jahre 2002 erstmalig verwendet. Damalige Internetangebote entwickelten sich zunehmend vom reinen statischen Wiedergeben von Inhalten zu vom Nutzer manipulierbaren dynamischen Inhalten, wodurch der Begriff *Web 2.0* entstand. Der Begriff RIA bezeichnet ein Konzept und basiert meist auf mehreren Web-Technologien wie zum Beispiel HTML, CSS, JavaScript und AJAX oder Flash und PHP. Um als RIA zu gelten, muss eine Anwendung über das Internet verfügbar sein. Allerdings muss sie nicht zwangsläufig im Browser ausgeführt werden, sondern kann auch als Desktopanwendung zum Einsatz kommen. RIAs im heutigen Sinn bieten dem Anwender Möglichkeiten, die früher Desktopanwendungen zur Verfügungen stellten. Die Individualisierung und Vereinfachung der Internetnutzung spielt eine entscheidende Rolle bei der Entwicklung neuer Anwendungen. Unter Interaktion versteht man beispielsweise das Ändern der Gestaltung, das Erzeugen und Verwalten von Daten, Drag&Drop-Funktionalitäten, sowie die Verwendung von Tastenkürzeln.

3. HTML5

3.1. Was ist HTML5?

3.2. Die Geschichte von HTML

HTML, "Hypertext Markup Language", ist die allgegenwärtige Auszeichnungssprache des World Wide Webs. Durch den geschickten Einsatz der paar Elemente (fortan: Tags, Wörter umgeben von eckigen Klammern z.B. <html>) die die Sprache unterstützt, konnten erstaunlich viele unterschiedliche Netzwerke von verlinkten Dokumente erstellt werden. Von bekannten Seiten wie Amazon, Ebay und Wikipedia bis hin zu personalisierten Blogs oder Webseiten, die sich zum Beispiel auf Katzenbilder spezialisiert haben. HTML5 ist die aktuellste Version dieser Auszeichnungssprache. Obwohl diese Version die bisher umfangreichsten Änderungen mit sich bringt, ist es nicht die erste Aktualisierung von HTML.

Sir Tim Berners-Lee zeichnet sich für die Entwicklung von HTML und damit dem Beginn des Internets verantwortlich. 1991 veröffentlichte er ein Dokument mit dem Titel "HTML Tags", in dem er weniger als zwei Dutzend Elemente, die für das Schreiben von Webseiten genutzt werden konnten, vorschlug. Darunter auch einige Tags, die bis dato verwendet werden, wie zum Beispiel:

- Listen (<ul>, <ol>, <li>)
- Überschriften (<h1>, <h2>, ...)
- Paragraphen (<p>)
- Titel (<title>)
- Links (<a>)

Die Verwendung von Tags war nicht Berners-Lees eigene Errungenschaft. Tags wurden bereits in der SGML (Standard Generalized Markup Language) verwendet. Anstatt einen komplett neuen Standard zu erfinden, erkannte Berners-Lee die Vorteile, bereits existierende Standards weiterzuentwickeln - ein Trend der auch in der Entwicklung der neuen HTML5 Spezifikation erkennbar ist.

3.2.1. Von der IETF zur W3C: Der Weg zu HTML4

Die erste offizielle Spezifikation war HTML 2.0, veröffentlicht durch die IETF, die "Internet Engineering Task Force". Viele der neuen Features dieser Spezifikation basierten dabei auf bereits veröffentlichte Implementierungen. So bot der im Jahr 1994 marktführende Web Browser Mosaic Autoren von Webseiten bereits die Möglichkeit, Bilder in Dokumente mittels eines <img>-Tags einzubinden. Das <img>-Tag wurde in die HTML 2.0 Spezifikation inkludiert. Die IETF wurde

allmählich durch die W3C, dem "World Wide Web Consortium", ersetzt. Folgende Aktualisierungen des HTML Standards wurden auf `http://www.w3.org` veröffentlicht. In der zweiten Hälfte der Neunziger wurde der Standard mehrmals überarbeitet bis 1999 HTML 4.01 veröffentlicht wurde.

3.2.2. XHTML 1: Die Vermischung von HTML und XML

Das nächste Update von HTML 4.01 trug den Namen XHTML 1.0. Das X steht dabei für "eXtensible". Die Spezifikation von XHTML 1.0 war ident zu der von HTML 4.01, somit wurden keine neuen Elemente und Attribute eingeführt. Die Spezifikationen unterschieden sich nur durch die zu verwendende Syntax zum Schreiben von Dokumenten. Während Autoren von HTML Dokumenten kaum Einschränkungen im Schreibstil von Elementen und Attributen hatten, setzte XHTML es voraus, dass die Regeln von XML, "Extensible Markup Language", einer strikteren Auszeichnungssprache auf der viele Technologien des W3C basierten, eingehalten werden. Aufgrund der strikteren Regeln einigten sich Autoren auf einen gängigen Schreibstil, der auch unter dem HTML 4.01 Standard Verwendung fand. Während Tags früher in Kleinbuchstaben, Großbuchstaben oder einer Mischung aus beidem geschrieben werden konnten, verlangte ein valides XHTML Dokument, dass alle Elemente und Attribute klein geschrieben werden. Während XHTML 1.0 noch auf HTML basierte und lediglich die strikteren Regeln von XML verwendete, war die XHTML 1.1 Spezifikation reines XML. Dieser Umstand führte zu schwerwiegenden Problemen. XHTML 1.1 Dokumente konnten nicht mehr unter dem Mime-Type *text/html* definiert werden. Der bis dato bekannteste Webbrowser Internet Explorer konnte jedoch Dokumente die mit einem XML Mime-Type publiziert wurden, nicht darstellen.

3.2.3. XHTML 2

Das W3C war mit der vierten Version von HTML der Meinung, dass der HTML basierte Ansatz seinen Zenith erreicht hat und setzte für die zukünftige Version vollständig auf XML. Trotz der fast identen Namen von XHTML 1 und XHTML 2, konnten die Unterschiede zwischen den beiden Spezifikationen nicht größer sein. Anders als XHTML 1, war XHTML 2 nicht abwärtskompatibel mit bereits existierenden Webinhalten oder vorangegangenen HTML Versionen. Es sollte ein vollkommen neuer Standard werden und es zeigte sich, dass dieser Weg nicht mit Erfolg gekrönt sein würde.

3.2.4. Die Spaltung: WHATWG

Innerhalb des W3C bildete sich eine Gruppierung die gegen die Ansätze rebellierten. Namenhafte Vertreter von Opera, Apple und Mozilla waren mit der eingeschlagenen Entwicklungrichtung des W3Cs nicht zufrieden. Ihnen war es wichtiger, mehr Aufmerksamkeit auf Formate zur Entwicklung von Webapplikationen zu richten. Bei einem Workshop in 2004 schlug Ian Hickson, der zu der Zeit bei Opera arbeitete, eine Weiterentwicklung von HTML vor, mit der es möglich sein soll, Anwendungen zu entwickeln. Der Vorschlag wurde durch die W3C abgelehnt. Unzufrieden mit der Entscheidung bildeten die Rebellen eine eigene Gruppe: Die "Web Hypertext Application Technology Working Group", oder kurz WHATWG.

3.2.5. Der Weg von Web Apps 1.0 zu HTML5

Von Beginn an operierte die WHATWG anders als das W3C. Während das W3C einen kosensorientierten Ansatz anwendet, Themen werden vorgetragen, es wird diskutiert und abgestimmt, wird

auch bei der WHATWG diskutiert und abgestimmt, allerdings liegt die finale Entscheidung, was in die Spezifikation komm und was nicht, beim Editor. Der Editor ist Ian Hickson. Der W3C Ansatz klingt demokratisch und fair, allerdings führt dieser auch dazu, dass der Prozess stark verlangsamt wird. Bei der WHATWG hat jeder die Möglichkeit mitzuwirken, da die letzte Entscheidung aber beim Editor liegt, entwickelt sich alles schneller. Schon zu Beginn der Gründung von WHATWG wurden die Projekte in zwei große Spezifikationen aufgeteilt: Web Forms 2.0 und Web Apps 1.0. Beide Spezifikationen sollen die bisherige HTML Spezifikation erweitern. Mit fortschreitender Entwicklung wurden alle Spezifikationen in eine einzige implementiert und in HTML5 umbenannt.

3.2.6. Die Wiedervereinigung

Während HTML5 von der WHATWG weiter entwickelt wurde, setzte das W3C die Entwicklung der XHTML 2 Spezifikation fort. Erst im Oktober 2006, schrieb Sir Tim Berners-Lee einen Blog-Post, in dem er zugab, dass der Versuch, das Internet von HTML auf XML zu übertragen, nicht funktionieren wird. Nur wenige Monate später entschied sich das W3C eine neue HTML Working Group zu bilden. Anstatt komplett von Anfang an zu beginnen, nutzen sie glücklicherweise die bisherige Arbeit der WHATWG als Basis für zukünftige HTML Spezifikationen. Dieser Umstand führte zu einer schwer überschaubaren Situation. Das W3C arbeitete parallel an den beiden unterschiedlichen, nicht kompatiblen Spezifikationen: XHTML 2 und HTML 5 (mit Leerzeichen). Währenddessen arbeitete die WHATWG, an der Spezifikation für HTML5 (ohne Leerzeichen) auf der die Arbeit der W3C aufbaut.

3.2.7. HTML5: 2012 und 2022

Heute ist der aktuelle Stand der HTML5 Spezifikation nicht mehr so undurchsichtig wie er früher war, allerdings gibt es noch immer offene Fragen.

Es gibt noch immer zwei Gruppen die an HTML5 arbeiten. Die wohl wichtigste Frage für Webentwickler ist: Wann können wir es nutzen? In einem Interview gab Ian Hickson an, dass die HTML5 Spezifikation frühestens 2022 den "proposed recommendation" Status erreichen wird. Klingt nach einer langen Wartezeit, allerdings bedeutet "proposed recommendation", dass die HTML5 Spezifikation zwei mal komplett implementiert werden muss. Als Vergleich: HTML 4 existiert nun seit über einem Jahrzehnt und hat noch nicht die gesetzten Features erreicht. Wenn man nun den Umfang der Spezifikation betrachtet, klingt dieses Datum hoch gesteckt. Browserhersteller sind nach wie vor nicht dafür bekannt, dass existierende Standards so schnell wie möglich implementiert werden. Der Internet Explorer benötigte mehr als ein Jahrzehnt um das *abbr*- Element richtig darstellen zu können.

Für Webentwickler war das Jahr 2012 wesentlich wichtiger. 2012 erreichte die HTML5 Spezifikation den "candidate recommendation" Status, der gleichbedeutend ist mit „fertig und abgeschlossen". Allerdings reicht das alleine leider auch nicht aus. Wirklich entscheidend ist die Frage: Ab wann unterstützen Webbrowser den neuen Standard? Schon die Veröffentlichung des CSS 2.1 Standards zeigte, dass man nicht auf die Fertigstellung der Spezifikationen warten sollte, sondern, wenn möglich, die Features nutzen sollte sobald es möglich ist. Das selbe gilt auch für HTML5. Sobald Webbrowser bestimmte Features der Spezifikation unterstützen, können diese auch jederzeit verwendet werden. Man darf nicht vergessen, dass HTML5 keine komplett neu entwickelte Sprache ist. Im Sinne der HTML Spezifikation ist es eher eine Evolution als eine

Revolution. Da HTML5 auf den früheren Versionen aufsetzt und immer irgendeine Version des HTML Standards zur Erstellung von Webseiten genutzt wird, wird bereits HTML5 genutzt.

3.3. Features

Mit HTML5 wird Entwicklern eine Fülle an neuen Funktionen und Features angeboten. An dieser Stelle soll ein Überblick über die Funktionalitäten gegeben werden, die diese Technologie auszeichnet.

3.3.1. Grafiken und Animation

- 2D Zeichentools
- Animationstools
- 3D Unterstützung
- 3D Transformationen
- SVG (Scaleable Vector Graphics)
- Skalierbare Inhalte
- Web-Fonts
- Filtereffekte (z.B. Weichzeichnen)
- Präsentationen

Die Spezifikation von HTML5 enthält bereits umfangreiche Ansätze zum Erstellen von Grafiken und Animationen. So können mit CSS3 und dem neuen Canvas-Element einfache und auch komplexe Grafiken erstellt und, mit dem Einsatz von der von dem Canvas-Element zur Verfügung gestellten Schnittstellen, über JavaScript animiert werden. Dank CSS3 können auch erstmals ohne Abstriche Schriftarten integriert und verwendet werden. Ein Feature das bis zu diesem Zeitpunkt nur von Flash angeboten wurde. Mit der Integration des SVG-Formats bietet auch HTML5 eine Möglichkeit um skalierbare Inhalte in Webseiten einzubinden. Einfache Filtereffekte können nun unkompliziert mit CSS3-Regeln erstellt werden. Für komplexe Effekte kann das Canvas-Element in Kombination mit JavaScript verwendet werden.

3.3.2. Schnittstellen (APIs)

- Zugriff auf das Filesystem
- Sockets
- Geolocation
- Datenaustausch

HTML5 bietet wie Adobe Flash eine Vielzahl an Schnittstellen, um mit anderen Systemen zu interagieren. Beispielsweise kann auf das Filesystem des Computers zugeriffen werden, um Daten zu speichern oder zu laden. Mittlerweile gibt es eine Unmenge an JavaScript Frameworks aus denen ein Entwickler zur Umsetzung von Projekten zurückgreifen kann. Das wohl bekannteste Framework ist jQuery. Frameworks helfen Entwicklern sonst schwer zu realisierende Features einfacher umzusetzen, so bietet jQuery mehrere Schnittstellen mit der ein Zugriff auf verschiedene Hardwarekomponenten, wie zum Beispiel die zur Standordbestimmung, möglich ist. Schon frühere Versionen von HTML unterstützten den Austausch von Daten über verschiedene Formate wie XML, JSON oder andere.

3.3.3. Multimedia

- Videounterstützung
- Audiounterstützung
- Spracheingaben

Eine der größten Neuerungen und Grund für den Schlagabtausch zwischen HTML5 und Adobe Flash bildet die Integration von nativen Audio- und Videoelementen und deren Unterstützung. Beide Elemente können dabei über seperate Schnittstellen angesprochen und gesteuert werden. Damit bietet sich eine Möglichkeit an Multimedia-Inhalte in Webseiten einzubinden, ohne auf den Adobe Flash Player zurückgreifen zu müssen. Einziger Wehrmutstropfen ist die noch zum Teil unvollständige Unterstützung durch bestehende Browser, jedoch sollten zukünftige Aktualisierungen dieses Problem beseitigen. In Kombination mit weiteren Neuerungen von HTML5 und JavaScript lassen sich interessante und komplexe multimediale Anwendungen erstellen. So kann Aufgrund der neuen Spracheingabeschnittstelle auf das Mikrofon des Computers zugegriffen und Spracheingaben verarbeitet werden.

3.3.4. Sonstiges

- Kompatibilität und Unterstützung auf unterschiedlichen Plattformen und Endgeräten
- Ansätze einer Programmiersprache

HTML5 Inhalte lassen sich auf jeder und für jede Plattform erstellen. Die Kompatibilität herzustellen liegt dabei nicht in der Hand des Entwicklers, sondern in denen der unterschiedlichen Browserhersteller. Durch die steigende Anzahl an Schnittstellen und den vielen Neuerungen nimmt JavaScript immer mehr die Form einer komplexen Programmiersprache an, mit deren Hilfe komplexe Anwendungen realisiert werden können.

3.3.5. HTML5 Spezifisch

- Semantisches Markup
- Suchmaschinenoptimierung

Semantisches Markup ist wohl das Feature das HTML im allgemeinen auszeichnet. Dank der angewandten Semantik ist es möglich, Inhalte mit, für den Computer verständliche, Bedeutung zu versehen. Ein Feature das sich bereits im Bereich der Suchmaschinenoptimierung als essenziel

erwiesen hat und mit der steigenden Bedeutung von Suchmaschinen nicht zu unterschätzen ist. Einige interessante Neuerungen von HTML5 umfassen die Erweiterungen der bestehenden Tags mit noch aussagekräftigeren Elementen.

3.4. Aktueller Einsatz

Keine andere Technologie konnte bis jetzt für einen ähnlich großen Hype sorgen wie HTML5. Von vielen wird die neue Spezifikation als die einzige Technologie angesehen, die notwendig sein wird, um in Zunkunft professionelle, anspruchsvolle und interaktive Webinhalte zu erstellen. Gegenwärtig werden diese Erwartungen allerdings noch nicht vollständig erfüllt. Die aktuelle Version, die veröffentlicht wurde, ist lediglich ein Entwurf für den finalen Webstandard. Bis dieser Standard verabschiedet wird, dauert es noch bis ins Jahr 2022. Das hat zur Folge, dass Entwickler mit Änderungen im Funktionsumfang und der bestehenden Features rechnen können.

Nichts desto trotz kann und soll der neue Standard für die Entwicklung von Webseiten und -applikationen genutzt werden. Mit HTML5 wurden viele Altlasten älterer HTML Versionen beseitigt und einige neue und zukunftsichere Funktionen hinzugefügt. In einigen Bereichen wie Multimedia, Interaktivität und Schnittstellen steht HTML5 dem bisherigen Platzhirsch Adobe Flash in den erwähnten Bereichen kaum noch nach. Im direkten Vergleich zu Adobe Flash löst HTML5 einige Aufgaben deutlich besser und verbraucht dabei auch noch weniger Ressourcen und Bandbreite.

Mit der kürzlichen Umstellung großer Videoportalen (beispielsweise Youtube) von Adobe Flash auf HTML5 lässt sich eine eindeutige Bewegung feststellen. Allerdings wird bis jetzt die HTML5 Variante lediglich als zusätzliche Möglichkeit neben Flash angeboten. In Zukunft soll jedoch komplett auf die Verwendung von Flash verzichtet werden. Was die Umstellung bei Webseiten betrifft, geht diese wesentlich gemächlicher voran.

Im Gegensatz zu Adobe Flash ist für die Verbreitung von HTML5 nicht nur der Einsatz des Standards sondern auch die Unterstützung der Features durch die verschiedenen Browser notwendig. Leider zeigt sich hierbei eine der Schwächen von HTML5: Längst nicht alle Features werden von allen Browsern unterstützt. Ganz im Gegenteil, interessante neue Funktionen werden von vielen Browsern nur teilweise oder garnicht unterstützt. Das führt unweigerlich zu sehr unterschiedlichen Darstellungen einer Webseite auf mehreren Browsern und damit zu einem Mehraufwand und höheren Kosten in deren Entwicklung und Testung.

3.5. Technologieverbund

Wenn von HTML5 gesprochen wird, ist in den häufigsten Fällen der Verbund aus der Markupsprache HTML in der Version 5, der deklarativen Sprache CSS in der Version 3 und der Skriptsprache JavaScript (auch bekannt als ECMAScript) gemeint. Alle drei Technologien greifen ineinander und bieten in ihrer Gesamtheit die neuen Features von HTML5.

3.5.1. HTML5

HTML ermöglicht es, Inhalte im Web zu strukturieren und darzustellen. Zusätzlich bietet HTML Funktionen um Texte, Bilder, Hyperlinks, Videos, Audio und andere multimediale Inhalte semantisch auszuzeichnen und damit für andere Systeme, zum Bespiel dem Computer, verständlich zu machen. Jegliche Inhalte, die im Web dargestellt werden sollen, müssen in die HTML-Struktur eingebunden werden, selbst solche, die zur Darstellung Plug-Ins von anderen Anbietern wie beispielsweise Adobe Flash benötigen.

Die gewünschten Inhalte werden dabei in verschiedene HTML-Elemente eingebettet. Jedes dieser HTML-Elemente erfüllt einen bestimmten Zweck und dieser reicht von Überschriften und Paragraphen bis hin zu Hyperlinks und Bildern. Zusätzlich können diese Elemente durch verschiedene Attribute noch genauer ausgezeichnet werden. Generische Elemente wie das div-Element, ergänzen den begrenzten Elemente-Umfang, um nicht eindeutig zuordenbare Inhalte auf einer Webseite zu platzieren. Über diesen Weg kann jede Form von Inhalt erstellt werden.

Für HTML5 wurden verschiedene Ziele festgelegt die es zu erreichen galt. Diese Ziele dienten zusätzlich dem Zweck, den neuen Standard konkurrenzfähig und zukunftssicher zu machen. Die Ziele waren:

Kompatibilität: Bereits bestehende Inhalte müssen weiterhin verwendet werden können. Neue Funktionen des Standards dürfen die bestehenden Inhalte nicht verändern oder beeinflussen.

Verwendbarkeit: Neue Funktionen sollen bestehende Probleme lösen und die Entwicklung einfacher machen.

Sicherheit: Ein wichtiger Aspekt, der im Web immer mehr an Bedeutung gewinnt.

Konsistenz: Sowohl in HTML als auch in XHTML sollen XML-Elemente genutzt werden können. Die Sprachen greifen dabei auf eine gemeinsame Dokumentation zurück.

Vereinfachung: HTML soll deutlich vereinfacht und durch konkrete Fehlermeldungen schneller und einfacher korrigierbar werden.

Universalität: Inhalte sollen in jeder Sprache und auf allen Endgeräten sowie Systemen darstellbar sein.

Zugänglichkeit: Inhalte und Funktionen sollen jedem zugänglich sein. Dies entspricht dem OpenSource-Gedanken.

Bis dato wurden bereits einige Ziele erreicht. Bis zur offiziellen Verabschiedung des Standards im Jahr 2022 werden mit Sicherheit alle Ziele erreicht werden.

3.5.2. CSS3

Mittels "Cascading Stylesheets", oder kurz CSS, wird die Darstellung von Elementen in einem HTML Dokument kontrolliert. So ist es möglich die Positionierung und Attribute wie Farbe, Schattierung oder ähnliches, von Elementen exakt zu definieren. CSS ist damit das Werkzeug der Wahl um Designvorgaben möglichst genau umzusetzen.

Die Trennung von HTML und CSS symbolisiert die Trennung von Inhalt und Design. Dies

ermöglicht eine unterschiedliche Darstellung der selben Inhalte durch unterschiedliche Stylesheets für verschiedene Endgeräte oder den Druck, ohne dabei den Inhalt selbst anpassen zu müssen.

Der Name Cascading Stylesheets ist abgeleitet von den zur Verfügung stehenden Vererbungsregeln, die das Format zulässt. Durch die Vererbungsregeln ist es möglich, ein große Auswahl an Elementen gleich aussehen zu lassen und dennoch jedes beliebige Element individuell anzupassen. Die von HTML gelieferten Attribute ID und Class dienen zur Unterscheidung von einmalig auftretenden Elementen (id) und ganzen Gruppierungen (class).

Mit dem Level 3 des Cascading Stylesheet Standards kommen viele neue Eigenschaften hinzu, die die Gestaltung von Elementen noch schneller und einfacher machen. Mit CSS2 konnten vorwiegend einfache Darstellungsparameter wie Farbe, Rahmen, Innenabstand und Aussenabstand sowie die genau Positionierung im Dokumentenfluss definiert werden. CSS3 ermöglicht nun etliche grafische Effekte wie Verläufe, Schatten oder abgerundete Ecken. Zusätzlich zählen zu den wichtigsten neuen Funktionen von CSS3 die Erstellung von einfachen Animationen, das Einbinden und Modifizieren von Schriftarten sowie Transformationen und Skalierungen von Inhalten.

Seit dem Jahr 2000 ist der CSS3 Standard in Entwicklung und die komplette Unterstützung durch Browserhersteller schreitet immer weiter voran. Vor allem durch das Aufkommen von HTML5 werden die Eigenschaften immer stärker in den Mittelpunkt gerückt. Aus diesem Grund werden die Funktionen von CSS3 oft dem HTML5 Standard zugeordnet.

3.5.3. JavaScript

Die Skriptsprache JavaScript übernimmt in dem Technologieverbund die logische Ebene. Die Verwendung von HTML als eigenständige Technologie bietet nur die statische Erstellung und Darstellung von Inhalten. Durch JavaScript lassen sich HTML-Elemente allerdings dynamisch verändern. Ein Trend der sich bei der Entwicklung von modernen Webseiten durchgesetzt hat und mittlerweile nicht mehr weg zu denken ist. JavaScript wurde 1995 offiziell von Brendan Eich eingeführt und ist aktuell in der Version 1.8.5 verfügbar. Allerdings unterstützen ein Großteil lediglich die Version 1.5 vollständig.

Die Skriptsprache ermöglicht die Manipulation von HTML-Elementen auf der im Browser aufgerufenen Seite, das dynamische Erstellen neuer Elemente oder das dynamische Nachladen von Inhalten mit Hilfe von XML und AJAX. Besonders für einfache Effekte wie ein Dropdown-Menü oder das Einblenden von Tooltips bietet sich der Einsatz von JavaScript an.

Als dritte Säule der Webseitenerstellung kümmert sich JavaScript daher vor allem um die Dynamik und Interaktivität von modernen Webseiten. Diese Funktionalitäten werden heutzutage von Nutzern erwartet. Die steigenden Erwartungen und die immer komplexer werdenden Webseiten fordern auch, dass JavaScript diesen gewachsen ist. Diese Anforderungen und die fehlende Unterstützung der aktuellsten Skript-Versionen machen die Entwicklung unverhältnismäßig schwer, saubere Lösungen für an sich einfache Probleme zu entwickeln. Für diesen Fall kommen die sogenannten JavaScript Frameworks ins Spiel. Frameworks bieten Anhand von kompatibler JavaScript Versionen neue Funktionalitäten. Besonders um einfach, sicher und effizient Animationen oder Effekte zu realisieren oder Elemente zu manipulieren bietet sich die Nutzung von Frameworks an. Einige der bekanntesten Frameworks sind dabei Prototype, MooTools und das

mit Abstand am häufigsten verwendete Framework jQuery.

Nur durch das Zusammenspiel des Grundgerüsts HTML, des Designs CSS3 und der Interaktivität durch JavaScript wird HTML5 zu dem Gesamtprodukt, welches in Zukunft die Entwicklung des Webs prägen wird und zu einem ernst zu nehmenden Konkurrenten für bereits etablierte Softwarelösungen wie Adobe Flash werden lässt.

3.6. Notwendigkeit von HTML5

Die letzte Version von HTML wurde gut 10 Jahre lang eingesetzt. Mit dem unglaublichen schnellen Wachstum im Bereich der mobilen Endgeräte, wie Smartphones und Tablets, und den steigenden Erwartungen der Nutzer im Bezug auf Inhalte und Funktionalitäten konnte die bestehende Version den gegebenen Ansprüchen nicht mehr gerecht werden. Um sowohl die geforderten Anforderungen zu erfüllen und zukunftssicher zu sein ist eine neue Version der Sprache notwendig. HTML5 kann diese Version werden.

Die bisher vergleichweise rasante Entwicklung von HTML5 wird sich daher auch in Zukunft kaum verlangsamen. Anders als bei einer proprietären Software beteiligen sich viele der führenden Unternehmen im Bereich Internet an der Entwicklung des neuen Standards. Damit kann man von einer umfassenden Unterstützung durch verschiedene Browser, Endgeräte und Schnittstellen ausgehen.

Um technische Neuerungen und moderne Webseiten zu gestalten mussten bisher immer die Funktionen von Adobe Flash oder ähnlicher Software genutzt werden. Die neuen Features von HTML5 sollen hingegen eine Möglichkeit bieten komplett auf Software-Lösungen von Drittanbieteren zu verzichten und dabei einen einheitlichen Standard für alle Geräte und Plattformen zu schaffen.

Vor allem im Bereich der Multimediainhalte wie Videos, Audio, Animationen, 3D und Interaktion musste Adobe Flash genutzt werden. Zu beachten ist dennoch, dass Software von Drittanbietern neben technischen Vorteilen auch einige Nachteile mit sich bringen. Einer der größten Nachteile stellt die kaum oder nur mit sehr viel Aufwand verbundenen Suchmaschinen-Optimierung von Flash-Inhalten dar. Ein Problem, dass in einer Zeit in der Online-Marketing immer mehr an Bedeutung gewinnt, nicht zu unterschätzen ist. Weiters werden laufend neue Sicherheitsproblemem bei den angebotenen Plug-Ins, die notwendig sind um entsprechende Inhalte darzustellen, bekannt und nur langsam oder auch unzureichend behoben. Aufgrund der vielen Nachteile wurde der Ruf der Kritiker, von Adobe Flash und proprietärer Software, nach einer Alternative immer lauter.

Mit der Entscheidung von Apple Adoble Flash auf ihren mobilen Endgeräten nicht mehr zu unterstützen, führte außerdem dazu, dass schnell Alternativen entwickelt werden mussten. HTML5 stellt dabei eine optimale Lösung dar.

4. HTML5 Canvas

4.1. Einleitung

Das Canvas Element bildet eines der größten Teile der HTML5 Spezifikation. Es bietet eine API für das Zeichnen von 2D Objekten, wie Linien, Füllungen, Bilder oder Text. Ein Vergleich mit MS Paint ist sehr gut um sich die Möglichkeiten mit dem Canvas Element vorstellen zu können. Tatsächlich gibt es bereits mehrere Implementationen, basierend auf dem Canvas Element, die Applikationen wie MS Paint vollständig ersetzen können. Es geht bereits so weit, dass laufend neue Zeichenapplikationen entstehen, die sich darum bemühen, als vollwertige Vektor-Zeichenprogramm, ähnlich wie Adobe Illustrator, angesehen zu werden. Das bemerkenswerte daran ist, dass es auch einige Anwendungen gibt, die ebenfalls auf mobilen Endgeräten, wie iPhones oder Android Smartphones, funktionstüchtig sind.

4.2. Geschichte

Das Konzept des Canvas Elements wurde durch Apple eingeführt um in Mac OS X Webkit Widgets für das Dashboard zu erstellen. Vor der Einführung des Canvas Elements konnten Zeichen Schnittstellen im Browser nur dann genutzt werden, wenn Plugins wie Flash von Adobe, Scaleable Vector Graphics (SVG), die Vector Markup Language (VML), welche nur in Internet Explorer vorhanden war, oder diverse JavaScript Bibliotheken eingesetzt wurden.
Das Canvas Element ist eine Umgebung für das Erstellen von dynamischen Bildern. Das Element selbst ist dabei sehr einfach zu verwenden. Lediglich die Dimensionen des Canvas und eine id, über die das Canvas anschließend über JavaScript referenziert werden kann, müssen festgelegt werden.

```
1  <canvas id="myCanvas" width="360" height="240">
2  </canvas>
```

Alles was sich zwischen den sich öffnenden und schließenden <canvas> Tags befindet, wird nur Browsern angezeigt, die das Canvas Element nicht unterstützen.

```
1  <canvas id="myCanvas" width="360" height="240">
2    <p>Canvas is not supported by your browser.</p>
3  </canvas>
```

Um auf dem Canvas zeichnen zu können muss JavaScript verwendet werden. Dabei muss zunächst das entsprechende Canvas über seine id ausgewählt und anschließend der gewünschte Kontext definiert werden. Die Auswahl des Kontexts beeinflusst die Funktionen der genutzen API:

```
1  var canvas = document.getElementById("myCanvas");
2  var context = canvas.getContext("2d");
```

Beim Verfassen dieser Arbeit standen nur der 2D oder der WebGL (3D) Kontext zur Verfügung. Mit der Auswahl des gewünschten Kontexts ist das Canvas Element bereit um für das Zeichnen

von Bilder genutzt zu werden. Die 2D API verfügt über alle Funktionen die bei einem herkömmlichen, teils kostenpflichtigen, Grafikprogramm wie zum Beispiel Adobe Illustrator erwartet werden: Es können Linien, Konturen, gefüllte Flächen, Verläufe, Schatten, Formen und Bézier Kurven gezeichnet werden. Der wesentliche Unterschied zwischen dem Canvas Element und einem Grafikprogramm besteht allerdings darin, dass das Canvas Element kein grafisches User Interface, zum Erstellen von Bildern, zur Verfügung stellt, sondern jeder Pixel per JavaScript programmiert werden muss.

4.2.1. Mit Code zeichnen

Um die Farbe einer Kontur bzw. einer Linie zu definieren ist folgender Code notwendig:

```
1 | context.strokeStyle = "\#990000";
```

Alles was nun folgend auf dem Canvas gezeichnet wird, hätte eine rote Kontur. Um ein Rechteck mit Kontur und ohne Füllung auf das Canvas zu Zeichnen gibt es die strokeRect Methode und ist folgendermaßen definiert:

```
1 | strokeRect(left, top, width, height);
```

Um nun ein rotes Rechteck zeichnen zu können, das 20 Pixel vom linken Rand und 30 Pixel vom oberen Rand des Canvas entfernt ist, 100 Pixel breit und 50 Pixel hoch ist, ist folgender Code notwendig:

```
1 | context.strokeRect(20, 30, 100, 50);
```

Dabei handelt es sich noch um ein sehr einfaches Beispiel um das Canvas zum Zeichnen zu nutzen. Der 2D Kontext bietet eine sehr umfangreiche Auswahl an vordefinierten Methoden wie fillStyle, zum Festlegen von Füllungen, fillRect, um ein gefülltes Rechteck zu zeichnen, lineWidth, zum Festlegen der Linienstärke, shadowColor, zum Festlegen der Schattenfarbe, und viele mehr. Theoretisch betrachtet kann jedes Bild das mit einem Programm wie Adobe Illustrator erstellt werden kann, ebenfalls mit dem Canvas Element realisiert werden. Praktisch sieht das anders aus. Ein Versuch würde sich als äußerst arbeitsaufwendig und in einer Unmenge an JavaScript äussern. Abgesehen davon, wurde das Canvas Element nicht zur Reproduktion von aufwendigen Pixel- oder Vektorgrafiken entwickelt.

4.2.2. Canvas: Was bringts?

Im vorherigen Abschnitt wurde ein Beispiel, die Reproduktion von Bildern, genannt, wofür das Canvas Element nicht verwendet werden sollte. Die wirkliche Macht des Canvas Elements liegt in der Möglichkeit, dass dessen Inhalt jederzeit, aufgrund von zum Beipiel Nutzereingaben, verändert werden kann. Gerade das Reagieren auf Nutzereingaben ermöglicht die Erstellung von Anwendungen und Spiele die ursprünglich eine Technologie wie Adobe Flash benötigten.
Eines der ersten Vorzeigebeispiele für die Möglichkeiten des Canvas Elements wurde von Mozilla Labs entwickelt. Bespin: Ein Code Editor der innerhalb des Browserfensters läuft. Eine sehr eindrucksvolle und nützliche Applikation, allerdings von vielen Entwicklern als perfektes Beispiel betrachtet, wofür das Canvas Element nicht genutzt werden sollte.

5. Adobe Flash

5.1. Was ist Flash?

Während die Popularität von HTML5 immer weiter steigt, werden immer mehr Vergleiche mit „Flash“ durchgeführt. Hierbei handelt es sich um ein Produkt der Firma Adobe, die sich auch unter anderem für Software wie Photoshop oder Illustrator verantwortlich zeigen. Allerdings bezieht sich das Wort Flash in den meisten Fällen nicht auf die gleichnamige Entwicklungsumgebung der Firma, sondern auf den Flash Player, der das Abspielen von proprietären SWF-Dateien im Browser ermöglicht. Anfänglich war Flash ein Programm um Illustrationen und einfache Animationen zu erstellen und wurde 1997 von der Firma Macromedia, welche im Jahr 2005 von Adobe aufgekauft wurde, entwickelt.

5.2. Die Geschichte von Flash und ActionScript

Seit der Veröffentlichung von Flash im Jahr 1996 wurde Flash und ActionScript gemeinsam weiter entwickelt. Mittlerweile stellt die Kombination aus Design- und Animations-Tools in Flash und den interaktiven Möglichkeiten von ActionScript eine der mächtigsten, vielseitigsten und beliebtesten Entwicklungsumgebung dar. Allerdings war ActionScript zu Beginn eine sehr bescheidene Programmiersprache.

Die ersten drei Versionen von Flash boten noch keine Programmiertools an und interaktive Abläufe wurden durch Drag-and-Drop Optionen als Actions festgelegt. Lediglich die Navigation durch die Zeitleiste und das Erstellen von Links waren über diese Actions möglich.

Flash 4 war die erste Version, die die Möglichkeit bot, Code mittels einer einfachen Skriptsprache zu schreiben, welche inoffiziell bereits ActionScript genannt wurde. Mit Flash 5 entwickelte sich ActionScript weiter und wurde zur offiziellen Skriptsprache. Mit jeder folgenden Version von Flash wurden auch die Möglichkeiten von ActionScript erweitert. ActionScript ermöglichte die interaktive Kontrolle von Text, Animation, Sound, Video, Data und mehr. 2003 wurde ActionScript 2.0 vorgestellt. Die Möglichkeiten dieses Produktes waren vergleichbar mit objekt-orientierten Programmiersprachen wie Java oder C#.

Professionelle Programmierer interessierten sich immer mehr für ActionScript als Entwicklungstool, jedoch zeigte sich, dass trotz der vergleichbaren Möglichkeiten ActionScript in der Performance noch nicht mit seiner Konkurrenz mithalten konnte. Der Grund dafür war, dass jede Version von ActionScript auf der vorherigen aufbaute. Der Flash Player war ursprünglich nicht für aufwändige Anwendungen und komplexe Spiele konzipiert, dennoch begannen Entwickler den Flash Player dafür zu nutzen. Damit wurde es klar, dass eine neue Version von ActionScript von Beginn an neu entwickelt werden musste um den Erwartungen gerecht zu werden.

2006 wurde von Adobe ActionScript 3.0 vorgestellt, welches bedeutend mehr neue Funktionalitäten und eine starke Performancesteigerung mit sich brachte. Flash CS3 war die erste Version von Flash, die ActionScript 3.0 unterstützte. Flash CS4 fügte neue Funktionen zu ActionScript 3.0 hinzu, unter anderem neue 3D Möglichkeiten, Kontrollfunktionen für Animationen und ActionScript Klassen für Adobe AIR. Flash CS5 und Flash CS6 erweiterten die Funktionalitäten unter anderem mit der Unterstützung von Controller und anderen Geräten, inklusive Multitouch-Funktionen und Touch-Screen Geräten.

Die aktuelle Version des Adobe Flash Players (Version 11) wurde im Oktober 2011 veröffentlicht.

5.3. Features

Adobe Flash bietet für Entwickler eine Fülle an Funktionen und Features. An dieser Stelle soll ein Überblick über die bedeutsamsten Funktionalitäten gegeben werden.

5.3.1. Grafiken und Animation

- 2D Zeichentools
- Animationstools
- 3D Unterstützung
- 3D Transformationen
- SVG (Scaleable Vector Graphics)
- Skalierbare Inhalte
- Text-Engine
- Filtereffekte (z.B. Weichzeichnen)
- Präsentationen

Adobe Flash bietet Entwicklern eine große Auswahl an Werkzeugen zur Erstellung von Grafiken und Animationen. Die umfangreiche Entwicklungsumgebung ermöglicht mittels integrierter Tools wie den Zeichentools, der Zeitleiste oder der inversen Kinematik die einfache Erstellung von Grafiken und Animationen, ohne dabei ActionScript Code verwenden zu müssen. Auch Texte lassen sich innerhalb der Entwicklungsumgebung bearbeiten und modifizieren und mühelos in die fertige Anwendung integrieren. Vordefinierte Effekte können auf eine Anwendung bequem per Knopfdruck angewendet werden. Allerdings ist es auch möglich alle vordefinierten Anwendungen mittels ActionScript anzusprechen, selbst zu programmieren oder auch mit weiteren Funktionalitäten zu erweitern. Durch die Nutzung von ActionScript wird auch die Programmierung von komplexeren Anwendungen ermöglicht. Adobe legt großen Wert auf die bisherigen Möglichkeiten im 3D-Bereich und möchte diese noch weiter ausbauen.

5.3.2. Schnittstellen (APIs)

- Zugriff auf das Filesystem
- Web Sockets
- Geolocation
- Datenaustausch

Adobe Flash bietet eine Vielzahl an Schnittstellen an, mit denen Systeme und Daten angesprochen werden können. Mit dem Zugriff auf das Filesystem des Computers ist es möglich, Daten zu laden, zu bearbeiten und zu speichern. Der Zugriff auf die Hardware des Computers wird ebenfalls durch Schnittstellen ermöglicht. Dadurch kann zum Beispiel das Mikrofon, die Kamera oder die Sensoren zur Standortbestimmung per Geolocation des Endgeräts genutzt werden. Für den Austausch von Daten zwischen Anwendungen oder dem Computersystem bietet Adobe Flash den Entwicklern weitere Schnittstellen für XML und andere Dateiformate an.

5.3.3. Multimedia

- Videounterstützung
- Audiounterstützung
- Streaming

Adobe Flash bot als eine der ersten Technologien die Möglichkeit, Video- und Audio-Elemente in Anwendungen zu integrieren. Die Unterstützung von verschiedenen Kodierungsverfahren und Optionen ermöglichen die einfache und vollständige Kontrolle über die Qualität und Datenmenge. Als große Stärke von Adobe Flash zeichnet sich die einfachen Realisierung von direkten Live-Streams ab.

5.3.4. Sonstiges

- Kompatibilität und Unterstützung auf unterschiedlichen Plattformen und Endgeräten

Adobe Flash unterstützt alle gängigen Desktop-Betriebssysteme wie Apples MacOS, Microsoft Windows und Linux. Lediglich das mobile Betriebssystem Android kann eine vereinfachte Version des Adobe Flash Players nutzen und damit Inhalte vollständig wiedergeben. Für die nicht unterstützten mobilen Betriebssysteme wie z.B. iOS können Flash Inhalte in HTML5 umgewandelt werden. Allerdings ist das Ergebnis aufgrund der begrenzten technischen Möglichkeiten nicht ident mit der originalen Flash Version.

5.3.5. Flash Spezifisch

- ActionScript
- Eigene Entwicklungsumgebungen (Adobe Flash, Flex)

Die in Adobe Flash integrierte objektorientierte Programmiersprache ActionScript ermöglicht die Realisierung umfangreicher und komplexer Anwendungen. Adobe bietet mit Adobe Flash eine umfangreiche aber teure Entwicklungsumgebung mit grafischer Oberfläche und integrierten Tools sowie Syntaxhervorhebungen und Programmierhilfen für ActionScript Code. Mit Flex gibt es auch eine kostenfreie Entwicklungsumgebung mit Tools und Hilfen für die Entwicklung von Anwendungen mittels ActionScript, allerdings ohne grafische Oberfläche und Werkzeuge zur Erstellung und Bearbeitung von Grafiken und Animationen.

5.4. Aktueller Einsatz

Trotz propritärer Technologien hat Adobe es geschafft, ihren Flash Player auf 99% Prozent aller Systeme unterzubringen. Gründe dafür waren die Fähigkeiten von Flash, mit denen Ziele erreicht werden konnten, die mit Hilfe von offenen Standards nicht möglich waren. In den Anfangstagen von Flash wurde es vor allem für Introanimationen, welche auf Startseiten eingesetzt wurden, interaktive Navigationen, die auf die Aktionen des Users mit z.B. Animationen reagierten, oder für die allseits bekannten Werbebanner genutzt. Die Erstellung ähnlicher Funktionen mit standardkonformen Technologien wie HTML 4.01, JavaScript oder CSS, war im Gegensatz zu Flash, nur schwer oder überhaupt nicht realisierbar. Selbst heute werden komplette Webseiten, Spiele und vor allem Werbebanner mittels Flash realisiert. Ein weiterer Vorteil von Flash ist das einfache Erlernen der Programmiersprache und Bedienen der Entwicklungsumgebung, wodurch auch Neueinsteiger sehr schnell zu ansehnlichen Ergebnissen kommen.

Für sich betrachtet vermitteln die Verbreitung des Adobe Flash Player und die genannten Fakten den Eindruck, dass Flash die unangefochtene Nummer Eins unter den Webtechnologien sein muss. Jedoch gilt es zu beachten, dass diese Informationen aus einer Zeit stammen, in der der Wandel zu HTML5 gerade erst begonnen hat, was zur Folge hat, dass aktuelle Zahlen anders aussehen würden. Dennoch verdeutlichen die Verbreitung und die genannten Fakten was für eine starke Plattform Adobe Flash bietet und in Zunkunft weiterhin bieten wird.

HTML5 hat es schwer, Flash in sämtlichen Bereichen zu verdrängen. Einige Anbieter, unter anderem YouTube, haben bereits mit der Umstellung ihrer Inhalte begonnen. In Bereichen in denen die Features der HTML5 Spezifikation bereits in einem fortgeschrittenen Entwicklungsstadium sind, können dafür bereits genutzt werden. In anderen Bereichen wird Flash auch in Zukunft unumgänglich bleiben, da es Features enthält, die mittels HTML5 nur umständlich oder überhaupt nicht umsetzbar sind. Dies sind zum Beispiel die Bereiche des gesicherten Online-Streamings oder die Erstellung von Rich Internet Applications (RIA).

Ein großer Vorteil von Flash stellt die hohe Verbreitung des Adobe Flash Player dar. Wie es aus dem ersten Absatz des Kapitels ersichtlich ist, kann dieser auf fast jedem Computer gefunden werden. Anders als HTML5 ist man dabei nicht von den Browserherstellern abhängig, um neue Funktionen einzubauen und zu unterstützen. Mit der Installation des Adobe Flash Player, sind lediglich eigene Updates der Anwendungen für Neuerungen und Inhalte notwendig. Bei Betriebssystemen ohne Unterstützungen des Flash Players sieht das wiederum anders aus. Als Beispiel verzichtet Apple vollständig auf die Unterstützung von Flash auf sämtlichen mobilen Endgeräten.

Veröffentlicht als proprietäre Software, wurden seitdem große Teile von ActionScript 3.0 durch Adobe selbst, Drittanbietern oder OpenSource Projekte öffentlich zugänglich gemacht, wodurch es mittlerweile möglich ist, unabhängig von Adobe Produkten Flash Inhalte zu erstellen. Mit Entwicklungsumgebungen wie Flex und Add-Ons für Eclipse wird versucht, für Entwickler immer bessere Tools zur Erstellung von Flash Anwendungen anzubieten.

Trotz möglicher aufkommender Konkurrenten lässt sich, aufgrund der hohen Nutzerzahl und der vielen Vorteile, mit Sicherheit sagen, dass Flash weiterhin ein wichtiger Bestandteil des World Wide Webs bleiben wird.

5.5. ActionScript 3.0

ActionScript ist die von Adobe für Flash entwickelte Programmiersprache. Die aktuelle Version 3.0 ist eine vollwertige objektorientierte Programmiersprache.

Erstmals erschien die Sprache mit dem Flash Player 4 im Jahr 1999. Die volle Unterstützung kam erst im Jahr 2000. Die erste Version beinhaltete einfache Befehle, mit denen die Steuerung von Flash-Elementen möglich war. Nutzer hatten damit erstmals die Möglichkeit frei über die Interaktionen der Elemente zu entscheiden.

Version 2.0 von ActionScript erschien im Jahr 2003 mit dem Flash Player 7 und erweiterte die Steuerungsmöglichkeiten von Flash Inhalten deutlich. Auch die ersten Ansätze von objektorientierter Programmierung und Vererbung von Klassen wurden implementiert. Im Vergleich zu anderen Skriptsprachen wie JavaScript war der Funktionsmöglichkeiten von ActionScript zu diesem Zeitpunkt bereits umfangreicher. So konnten Entwickler bereits komplexe multimediale Inhalte erstellen, die vom Nutzer, dank hoher Interaktivität, welche mittels ActionScript ermöglicht wurde, gesteuert werden konnten.

2006 erschien die bis heute aktuelle Version von ActionScript: 3.0. Mit der Version 3.0 wurden auch die meisten Neuerungen eingeführt. So unterstützt ActionScript 3.0 die von anderen Programmiersprachen, wie z.B. Java oder C, bekannten Paradigmen der klassenbasierten Objektorientierung und Typisierung zur Laufzeit vollständig. Mittels einer virtuellen Maschine, in der ActionScript Code nun ausgeführt wird, wird garantiert, dass der Code auf jeder Plattform gleich ausgeführt wird. Im Hinblick auf mobile Plattformen und deren starkem Wachstum ein zukunftssicherer Schritt.

Entwicklern werden verschiedene Bibliotheken von Adobe mit den Funktionalitäten von Flash angeboten, sodass auch außerhalb ohne der Nutzung der offiziellen Entwicklungsumgebung Code generiert werden kann. So kann auch mit wenig Mitteln komplexere Projekte auf der Basis von ActionScript realisieren, die weit über den Einsatz im Flash Player als Browser Plug-In hinaus gehen. So hat das Aufkommen von ActionScript 3.0 besonders in den letzten Jahren dafür gesorgt, dass sich Flash im Markt der Rich Internet Applications eine starke Position sichern konnte.

5.6. Apples Argumente gegen Flash

Eine oft verwendete Aussage und immer wiederkehrendes Streitthema in vielen Diskussionen ist sicherlich, ob HTML5 langfristig Flash als Webentwicklungswerkzeug ablöst. Apple spielt hierbei eine große Rolle. Das Unternehmen verzichtete seit der Einführung ihrer iDevices (iPhone, iPod touch, iPad) auf die Nutzung von Flash. Vor allem die fehlende Möglichkeit, Flash auf dem iPad zu verwenden, führte vermehrt zu Kritik. Der User sei eingeschränkt und ohne Flash handelt es sich nicht um das „echte Internet", welches Apple mit ihren Produkten anpreist. Der verstorbene Steve Jobs, CEO von Apple, hat daraufhin einen offenen Brief verfasst, der den Nutzern erklärt, warum die Firma auf Flash verzichtet hat und auch weiterhin darauf verzichten wird. Steve Jobs Hauptargumente waren und gelten noch bis heute:

1 Flash ist ein 100 Prozent proprietäres System von Adobe, welches trotz der allgemeinen Verfügbarkeit des Flash Players die Zügel in der Hand hat. Adobe könnte ganz alleine über die Richtung entscheiden, in die Flash geht oder die Preisgestaltung ändern. Apple möchte das Internetstandards offen sein sollen, selbst wenn es auf die eigenen Geräte zutrifft.

3 Flashprodukte sind laut einer Analyse des Softwarehauses Symantec mit am anfälligsten für Sicherheitsprobleme. Sie seien einer der Hauptgründe für Abstürze auf dem Mac, auch die Performance auf Mobiltelefonen würde zu wünschen übrig lassen.

4 Flash beansprucht die Akkulaufzeit auf mobilen Geräten, da das Dekodieren von Videomaterial über die Software laufen muss. H.264 würde über die Hardware dekodiert werden.

5 Flash wurde nicht für Touchscreens geschaffen. Viele Webseiten müssten ihre Seite komplett neu konzipieren und entwickeln. Entwickler sollten aber dann doch auf modernere Technologien wie HTML5, CSS3 und Javascript zurückgreifen.

Ein kommerzieller Hersteller wirft somit einem anderen kommerziellen Hersteller vor, zu proprietär zu sein - das mag durchaus merkwürdig erscheinen. Vor allem weil die Kunden von Apple auch im eigenen System eingezäunt und der Kontakt zu anderen Systemen über Schnittstellen so gering wie möglich gehalten wird und bei Adobe nur von der theoretischen, aber unwahrscheinlichen Möglichkeite ausgegangen wird, dass diese die Vormachtstellung ihres Browserplugins missbrauchen könnten.

Auch die Performanceprobleme scheinen bei Googles Smartphone Betriebssystem Android seit der Version 2.2 gelöst worden zu sein. Denn seitdem läuft Flash stabil und schnell, ohne den Akku zu sehr zu beanspruchen.

Die Behauptung das Flash nicht für Touchscreen ausgelegt worden ist, ist an und für sich wahr, das gleiche gilt allerdings auch für HTML. Auch hier gibt es eben so viele mausbasierende Aktionen, wie z.B. Rollover Animationen.

6. HTML5 Canvas und Adobe Flash: Vergleich und mögliche Auswirkungen

6.1. Vorraussetzungen

Sowohl Nutzer als auch Entwickler benötigen um HTML5 und Adobe Flash einsetzen zu können verschiedene Tools. Auf Entwicklerseite sind die Möglichkeiten sehr vielfältig, sodass eine vollständige Auflistung den Rahmen dieser Arbeit sprengen würde. Auf Nutzerseite ist die Auswahl der Tools ebenfalls breit gefächert, allerdings ist die Anzahl der notwendigen Programme vergleichweise geringer und kann daher genauer untersucht werden.

Die Entwicklung von Flash Anwendungen setzt einen Editor vorraus, der in der Lage ist ActionScript Code zu kompilieren. Dabei gibt es eine umfangreiche Auswahl an verschiedenen Editoren, die nicht zwingend von Adobe kommen müssen. Adobe bietet mit Adobe Flash CS6 eine Entwicklungsumgebung, welche sämtliche Funktionen von Flash und ActionScript unterstützt und dem Entwickler neben dem Editieren des ActionScript Codes eine grafische Oberfläche bietet, die es vereinfacht, Animationen zu erstellen oder Grafiken einzubinden, allerdings auch sehr kostspielig ist. Über die kostenlos erhältlichen Editoren können zwar alle Funktionen von ActionScript genutzt werden, aber es werden keine Tools zur Erstellung von Grafiken und Animationen angeboten.

Im Vergleich dazu benötigen HTML5 Entwickler lediglich einen Editor, mit dem textbasierte Inhalte erstellt werden können und einen Browser, der diese Inhalte korrekt darstellt. Die Lösungen hierfür sind nahezu grenzenlos, weshalb Entwickler aufgrund ihrer Präferenzen die für sie beste Lösung wählen können. Der komplette Aufbau der Webseite wird im Editor in der entsprechenden HTML5 Syntax geschrieben. Multimediale Inhalte lassen sich ebenfalls über das Markup einbinden, und können zuvor mit den entsprechenden Programmen den Wünschen angepasst werden. Adobe ist dabei eines der ersten Unternehmen, die Softwarelösungen bieten, mit denen Webseiten und Animationen in HTML5 mittels einer grafischen Oberfläche erstellt und bearbeitet werden können. In Zukunft ist es möglich, dass andere Unternehmen diese Idee aufgreifen und ähnliche Programme anbieten werden, sodass HTML5 im Bezug auf grafische Entwicklungsumgebungen Adobe Flash in nichts nachstehen wird.

Die technischen Vorraussetzungen für Nutzer von Flash-Inhalten sind relativ einfach zu erfüllen. Der Nutzer müssen lediglich eine aktuelle Version des Adobe Flash Players auf seinem System installiert haben, sei es als Browser Plug-In oder als Standalone Version, und das verwendete System muss die Hardware-Mindestanforderungen von Adobe erfüllen. Da diese Mindestanforderungen äusserst gering sind, sollten alle aktuellen Systeme, unabhängig ob mobil oder Desktop, in der Lage sein Flash-Inhalte abzuspielen. Da der Adobe Flash Player auf jedem gängigen System verwendet werden kann, ist auch die Entwicklung über mehrere Plattformen in der Regel kein Problem.

Für die Nutzung von HTML5 Inhalten wird lediglich ein Browser vorrausgesetzt, der diese korrekt darstellen kann. Der Entwickler hat dafür zu sorgen, dass jeder Browser die Seite soweit korrekt darstellt, sodass der Nutzer auf die Inhalte zugreifen kann. Das Problem hierbei ist, dass der Browsermarkt sehr stark fragmentiert ist und diese sowohl von Hersteller zu Hersteller als auch von Version zu Version eine unterschiedlich fortgeschrittene Unterstützung von HTML5 anbietet. Die Hardware- Mindestanforderungen sind dabei an den jeweiligen Browser gebunden. Allerdings sind auch diese sehr gering wodurch auch schon auf mobilen Systemen Browser integriert sind die eine sehr fortgeschrittene Unterstützung von HTML5 gewährleisten.

6.2. Betriebssysteme

HTML5 ist ein offener Standard. Das bedeutet, dass die Technologie nicht von einem großen Softwarehersteller entwickelt wird und somit für die Öffentlichkeit frei zugänglich ist. Das hat zur Folge, dass der Grad der Unterstützung von HTML5 abhängig von der Einbindung in den Browser der jeweiligen Browserhersteller ist. Jedes System mit einer aktuellen Version eines Browsers kann damit die Funktionen von HTML5 verwenden und darstellen.

Adobe Flash stellt das extreme Gegenteil zu HTML5 dar. Die Technologie ist eine von Adobe entwickelte proprietäre Software und setzt vorraus, dass für Browser ein Plug-In und für das System der Adobe Flash Player installiert sein muss, um Flash-Inhalte darstellen zu können. Damit hat Adobe die Macht über die Entscheidung der zu unterstützenden Plattformen. Sollte also Adobe sich dazu entscheiden, eine bestimmte Plattform nicht mehr zu unterstützen, haben die Nutzer dieser Plattform keine Möglichkeit mehr Flash-Inhalte zu konsumieren. Apple hat sich aus diesem Grund dazu entschieden, auf ihren mobilen Engeräten komplett auf die Unterstützung von Adobe Flash zu verzichten und ihre Aufmerksamkeit auf offene Standard wie HTML5, CSS3 und JavaScript zu richten.

Die gängigsten Desktop-Betriebssysteme (Windows, MacOS und Linux) unterstützen sowohl HTML5, mittels diverser Browser, und Adobe Flash, mittels Plug-In oder Player, und können damit Inhalte beider Technologien darstellen. Bei mobilen Betriebssysteme sind immer vorinstallierte Browser vorhanden, die einen Großteil der Features von HTML5 unterstützen. Für die Unterstützung von Adobe Flash müssen zusätzliche Applikationen installiert werden.

Um Flash-Inhalte auf Apples mobilen Endgeräte nutzen zu können, bietet Adobe eine Softwarelösung, wodurch Flash Inhalte in HTML5 Inhalte umgewandelt werden. Allerdings gilt es dabei zu beachten, dass die umgewandelten Inhalte nur selten den originalen Inhalten entsprechen, aufgrund der fehlenden Funktionen seitens HTML5 oder der hohen Komplexität der Inhalte. Adobe hat bereits angekündigt, die Entwicklung des Adobe Flash Players für mobile Plattformen einzustellen und sich auf Desktop Plattformen zu konzentrieren. Mit dieser Entscheidung seitens Adobe in Zukunft der mobile Bereich sicherlich HTML5 gehören.

6.3. Browserunterstützung

Wie im letzten Abschnitt bereits beschrieben, ist die Darstellung von HTML5-Inhalten abhängig von dem genutzten Browser. Unter den Browsern gibt es eine breite Auswahl, die alle eine unterschiedlich fortgeschrittene Unterstützung der Funktionen und Features von HTML5 bieten.

Das führt automatisch zu dem Problem, dass eine Webseite in unterschiedlichen Browsern unterschiedlich dargestellt wird. Wie weit die Darstellung dabei vom Original abweicht hängt von dem jeweiligen Browser ab. Das stellt eines der größten Probleme im Bereich der Webentwicklung mit den offenen Standards dar. Ein Entwickler muss sich darum bemühen, dass eine möglichst große Anzahl an Nutzern die angebotene Seite/Applikation optimal nutzen kann.

An dieser Stelle soll daher ein Überblick darüber gegeben werden, welche Browser welche Features und Funktionen von HTML5 unterstützen und der Vergleich aufgestellt werden, ob diese mit Adobe Flash auch erstellt/dargestellt werden können.

6.4. Leistungstests

Bei der Auswahl einer Technologie ist neben den gebotenen Funktionen, dem damit verbundenen Entwicklungs- und Lernaufwand und den Kosten auch die Leistungsfähigkeit auf verschiedenen Betriebssystemen und Endgeräten entscheidend. Speziell die Anzeigerate bei Videos und Animationen sowie die Belastung des Prozessors spielen eine tragende Rolle, da die Anzeigerate möglichst konstant und hoch und zeitgleich die Belastung des Prozessors möglichst gering sein soll.

Um diese Faktoren austesten zu können werden im Internet verschiedene Tests angeboten, die eine bestimmte Situation simulieren und die entsprechenden Ergebnisse liefern. Diese Tests, auch bekannt unter dem Namen "Benchmarks", decken dabei einen möglichst großen Testbereich ab. So reichen diese von der Darstellung von Grafiken über das Berechnen von Partikeln bis hin zu einem Belastungstest des Prozessors.

Die Benchmark Ergebnisse können dabei lediglich als Richtwert angesehen werden, da diese abhängig von dem beim Test verwendeten Betriebssystem und der verwendeten Software (Browserhersteller und Version bzw. Version des Flash Players) stark voneinander abweichen können.

In den folgenden Abschnitten werden verschiedene Benchmarks verwendet, um die Leistung von HTML5 und Adobe Flash möglichst genau zu vergleichen. Bei den Benchmarks wird darauf geachtet, dass Funktionen verwendet werden, die sowohl HTML5 als auch Adobe Flash zur Verfügung stellen damit für beide Technologien die gleichen Bedingungen bestehen.

Testumgebung

Um sämtliche am Markt verfügbaren Browser zu testen, werden zwei unterschiedliche Testsysteme verwendet. Die Testsysteme werden wie folgt spezifiziert:

- Gerät: Macbook Air (13 Zoll, Mitte 2012)
- Betriebssystem: OS X Version 10.9.1 (Mavericks)
- Prozessor: 2 GHz Intel Core i7
- Speicher: 8 GB 1600 MHz DDR3
- Grafikkarte: Intel HD Graphics 4000 1024 MB

- Festplatte: Flash-Speicher (SSD)

und

- Gerät: HP Compaq nc6400 (13 Zoll)
- Betriebssystem: Windows 7 Professional, Service Pack 1
- Prozessor: 2 GHz Intel Core Duo 2 T7200
- Speicher: 2 GB
- Grafikkarte: ATI Mobility Radeon X1300 256 MB
- Festplatte: HDD

Die Auswahl der Browser wird wie folgt spezifiziert:

- Firefox 26.0
- Chrome 32.0.1700.102
- Opera 19.0.1326.47
- Safari 7.0.1 (nur OS X)
- Internet Explorer 11 (nur Windows)

Die getestete Version vom Adobe Flash Player ist:

- Adobe Flash Player 12.0.0.38

6.4.1. GUIMark 2: The rise of HTML5

GUIMark 2 ist ein kostenloser Benchmark entwickelt von Sean Christmann für HTML5 und Adobe Flash. Christmann ist davon überzeugt, dass bei einer vollen Ausschöpfung der Rendering Pipelines einfacher entschieden werden kann, welche Technologie am Besten für interaktiven Inhalt im Web genutzt werden soll. Laut Christmann tendieren Entwickler hauptsächlich dazu die Nutzbarkeit der Technologien zu vergleichen, während in Wirklichkeit die meiste Prozessorleistung für interne Rendering Schnittstellen genutzt wird.

Der Benchmark ist in drei unterschiedliche Tests unterteilt: Vektor-, Bitmap- und Textdarstellung. Die Testfälle sind dabei möglichst realen Alltagsszenarien nachempfunden. Zusätzlich sind einige Testfälle vorhanden die ebenfalls für mobile Endgeräte entwickelt wurden, um die Aussagen von Steve Jobs, im Bezug auf die mobile Leistungsfähigkeit von HTML5 und Adobe Flash, zu hinterfragen.

Bei den folgenden Tests wird die Anzeigerate in Bilder pro Sekunde (fps, Frames per Second) gemessen und miteinander verglichen.

Vector Charting Test

Dieser Benchmark wurde konzipiert um die Funktionen der Vektor APIs mittels einem animierten Diagramm auszulasten. Das Diagramm wird dabei durch den massiven Gebrauch von Linien und aufwendigen Transparenzfüllungen erstellt.

	HTML5	Flash
OSX 10.9.1		
Firefox	6,86 fps	43,85 fps
Chrome	30,62 fps	48,38 fps
Opera	29,86 fps	46,98 fps
Safari	**55.76 fps**	**49.96 fps**
	i.D. 30,7 fps	i.D. 47,3 fps
Windows 7 SP1		
Firefox	10,05 fps	23,96 fps
Chrome	12,42 fps	**59,89 fps**
Opera	**13,52 fps**	15,26 fps
IE	10,24 fps	22,43 fps
	i.D. 11,6 fps	i.D. 30,39 fps

Tabelle 6.1.: Testergebnisse - Vector Charting Test

Auf dem Macbook Air zeigt sich, dass Apples Browser Safari mit rund 55fps die beste HTML5 Leistung bringt und die Leistung von Flash überall gleich konstant bleibt.

	HTML5
OSX 10.9.1	
Firefox	44,91 fps
Chrome	54,43 fps
Opera	54,68 fps
Safari	**56.92 fps**
Windows 7 SP1	
Firefox	10.54 fps
Chrome	27.36 fps
Opera	**30.38 fps**
IE	10.51 fps

Tabelle 6.2.: Testergebnisse - Vector Charting Test 1 Pixel Stroke

Bitmap Gaming Test

Der Bitmap Test simuliert ein Spiel auf dem Prinzip des Towerdefense. Dabei wird eine Vielzahl an Bitmaps pro Bild unterschiedlich animiert. Mit jedem fertigen Bild muss dieses auch wieder gelöscht werden um alle neuen Zustände im neuen Bild darstellen zu können. Um einen Tiefeneffekt zu erzeugen werden Bitmaps auf unterschiedlichen Ebenen positioniert (Z Depth Ordering) und um die Bitmaps zu skalieren wird Anti-alising verwendet.

	HTML5	Flash
OSX 10.9.1		
Firefox	20,9 fps	36,95 fps
Chrome	55,61 fps	35,02 fps
Opera	55,86 fps	36,28 fps
Safari	**57.03 fps**	**39.33 fps**
	i.D. 47,35 fps	i.D. 36,9 fps
Windows 7 SP1		
Firefox	7,34 fps	**20,09 fps**
Chrome	14,68 fps	17,55 fps
Opera	**18,71 fps**	18,79 fps
IE	10,66 fps	18,09 fps
	i.D. 12,85 fps	i.D. 18,63 fps

Tabelle 6.3.: Testergebnisse - Bitmap Gaming Test

Text Column Test

Bei diesem Test wird die Text Layout und Rendering Engine von HTML und Flash ausgelastet. Es werden Custom Fonts, die mit CSS3 eingeführt wurden, eingebunden und Multibyte Character Strings dargestellt. Anders als die vorherigen Benchmarks stellt dieser Testfall kein reales Testszenario dar, sondern dient der Analyse, wie viel Zeit benötigt wird, um eine textlastige Webseite vollständig zu laden. Christmann nennt diesen Test den „Eisberg"- Test, da ungefähr 80% der Prozessorbelastung ausserhalb des Renderviews (des sichtbaren Bereichs) entstehen. Dennoch soll der Test realistische Ergebnisse liefern, da beim Scrollen berechnet werden muss, wieviele Zeilen des Textes nun in bzw. aus dem Renderview bewegt werden müssen. Bei HTML Webseiten geschieht dies immer, sobald eine Seite geöffnet wird und die Auflösung des genutzen Monitors zu klein ist, um diese komplett anzeigen zu können.

	HTML5	Flash
OSX 10.9.1		
Firefox	28,09 fps	32,3 fps
Chrome	**31,85 fps**	**35,74 fps**
Opera	28,54 fps	30,08 fps
Safari	**31,85 fps**	34,8 fps
	i.D. 30,08 fps	i.D. 33,23 fps
Windows 7 PS1		
Firefox	**16,15 fps**	11,53 fps
Chrome	15,29 fps	**18,74 fps**
Opera	11,39 fps	11,38 fps
IE	10,6 fps	11,08 fps
	i.D. 13,36 fps	i.D. 13,18 fps

Tabelle 6.4.: Testergebnisse - Text Column Test

6.4.2. GUIMark 3: Mobile Showdown

6.4.3. The Man in Blue Animation Benchmark

6.4.4. Animation

6.4.5. Video

6.5. Besonderheiten

6.5.1. Barrierefreiheit

Barrierefreiheit im Internet bedeutet, Webseiten so zu gestalten, dass jeder Mensch unabhängig von seinen körperlichen Einschränkungen die Inhalte der Webseite nutzen kann. Dabei stellt besonders die Aufbereitung der Inhalte für Menschen mit Sehbehinderungen eine Schwierigkeit dar. Nichtsdestotrotz soll diese und auch andere Behinderungen bei der Aufbereitung von Inhalten für das Internet berücksichtigt werden.

Das W3C hat bereits im Jahre 1999 die erste Empfehlung für barrierefreies Internet ausgesprochen. Für das Erstellen barrierefreier Webseiten werden, seit 2008, Richtlinien in den Web Content Accessibility Guidlines (WCAG 2.0) festgehalten und der Öffentlichkeit zugänglich gemacht. Die WCAG 2.0 bilden nun einen flexibleren und testbareren, neuen Standard für barrierefreies Webdesign. Dieser deckt Zugänglichkeitsanforderungen für alle Arten von Web-Inhalten (Text, Bilder, Audio oder Video) sowie Web- Applikationen ab und definiert technologieunabhängige Richtlinien und Erfolgskriterien.

Dabei sind die WCAG 2.0 pyramidenartig aufgebaut und umfassen vier Ebenen:

1. 4 Prinzipien
2. 12 Richtlinien
3. 61 Erfolgskriterien
4. unzählige Techniken

Die ersten drei Ebenen stellen einen Maßstab und das Fundament der Richtlinien dar. Im Gegensatz umfasst die vierte Ebene ergänzende Dokumente, die nicht normativ sind und regelmäßig aktualisiert werden. Im folgenden werden die Prinzipen und Richtlinien genauer betrachtet:

1. Prinzip: Wahrnehmbar - Informationen und Bestandteile der Benutzerschnittstellen müssen den Benutzer so präsentiert werden, dass diese sie wahrnehmen können.
    - Richtlinie 1.1 Textalternativen: Stellen Sie Textalternativen für alle Nicht-Text-Inhalte zur Verfügung, so dass in andere vom Benutzer benötigte Formen geändert werden können, wie zum Beispiel Großschrift, Braille, Symbole oder einfachere Sprache.
    - Richtlinie 1.2 Zeitbasierte Medien: Stellen Sie Alternativen für zeitbasierte Medien zur Verfügung.
    - Richtlinie 1.3 Anpassbar: Erstellen Sie Inhalte, die auf verschiedene Arten dargestellt werden können (z.B. einfacheres Layout), ohne dass Informationen oder Struktur verloren gehen.

- Richtlinie 1.4 Unterscheidbar: Machen Sie es Benutzern leichter, Inhalt zu sehen und zu hören einschließlich der Trennung von Vorder- und Hintergrund.

2. Prinzip: Bedienbar - Bestandteile der Benutzerschnittstelle und Navigation müssen bedienbar sein.
   - Richtlinie 2.1 Per Tastatur zugänglich: Sorgen Sie dafür, dass alle Funktionalitäten per Tastatur zugänglich sind.
   - Richtlinie 2.2 Ausreichend Zeit: Geben Sie den Benutzern ausreichend Zeit, Inhalte zu lesen und zu benutzen.
   - Richtlinie 2.3 Anfälle: Gestalten Sie Inhalte nicht auf Arten, von denen bekannt ist, dass sie zu Anfällen führen.
   - Richtlinie 2.4 Navigierbar: Stellen Sie Mittel zur Verfügung, um Benutzer dabei zu unterstützen zu navigieren, Inhalte zu finden und zu bestimmen, wo sie sich befinden.
3. Prinzip: Verständlich - Informationen und Bedienung der Benutzerschnittstellen müssen verständlich sein.
   - Richtlinie 3.1 Lesbar: Machen Sie Inhalte lesbar und verständlich.
   - Richtlinie 3.2 Vorhersehbar: Sorgen Sie dafür, dass Webseiten vorhersehbar aussehen und funktionieren.
   - Richtlinie 3.3 Hilfestellung bei der Eingabe: Helfen Sie den Benutzern dabei, Fehler zu vermeiden und zu korrigieren.
4. Prinzip: Robust - Inhalte müssen robust genug sein, damit sie zuverlässig von einer großen Auswahl an Benutzeragenten einschließlich assistierender Techniken interpretiert werden können.
   - Richtlinie 4.1 Kompatibel: Maximieren Sie die Kompatibilität mit aktuellen und zukünftigen Benutzeragenten, einschließlich assistierender Techniken.

Im Vergleich HTML5 und Adobe Flash ist es verständlich, dass die Webtechnologie HTML einen Vorteil daraus zieht, dass die aufgezeigten Richtlinien vor allem für das Erstellen von klassischen Webseiten gegeben wurden. Allerdings bietet auch Adobe Flash verschiedene Möglichkeiten, um diese Richtlinien erfüllen zu können. Diese sind jedoch mit deutlich höherem Aufwand verbunden, als es bei HTML der Fall ist. Dieses Problem ist nicht nur auf Adobe Flash beschränkt, sondern umfasst auch andere im Web verwendete Technologien und Programmiersprachen, die bei der Umsetzung der Richtlinien für Barrierefreiheit Schwierigkeiten bereiten. Das hat zur Folge, dass zum Beispiel blinde Menschen große Schwierigkeiten bei der Nutzung von Nicht-HTML Inhalten haben, da diese nicht von z.B. Screenreadern verarbeitet werden können.

Mit den aktuellen Versionen des Adobe Flash Players und der ActionScript Version 3 bietet Adobe Entwicklern neue Möglichkeiten, um dem Nutzer alternative Inhalte anzubieten. Grundsätzlich sind die Möglichkeiten dabei ähnlich denen von HTML5. Damit wären barrierefreie Webseiten beziehungsweise Anwendungen durchaus auch mit Adobe Flash umsetzbar. Trotzdem bleibt die Umsetzung der Richtlinien des W3C ein schwieriges Unterfangen und ist deutlich aufwändiger und somit auch teurer, sodass häufig darauf verzichtet wird.

Es zeigt sich aus dem Vergleich HTML5 und Adobe Flash und bei der Betrachtung der

durch die W3C empfohlenen Richtlinien, dass Barrierefreiheit im Internet nicht eine Frage der eingesetzten Technologie/n sondern der vorhandenen Ressourcen (Zeit, Geld, Mitarbeiter, ...) ist.

6.5.2. Suchmaschinenoptimierung

Der Bereich der Suchmaschinenoptimierung hat sich in den letzten Jahren enorm entwickelt. Kaum eine Webseite, unabhängig ob privat oder kommerziell genutzt, kommt ohne sie aus. Das liegt vorwiegend daran, dass Suchmaschinen wie Google, Yahoo oder Bing dem Nutzer die Suche nach interessanten und relevanten Inhalte vereinfachen. Die Suchmaschinenoptimierung spielt dabei insofern eine gewichtige Rolle, dass für viele Nutzer nur die Suchergebnisse auf der ersten Seite relevant sind. Oft wird der Suchstring angepasst bevor überhaupt Ergebnisse auf der zweiten oder dritten Ergebnisseite in Betracht gezogen werden. Somit ist es für den Betreiber einer Webseite wichtig, dass seine Webseite möglichst weit vorne im Suchmaschinenindex auftaucht.

Inhalte die mit Adobe Flash erstellt wurden, haben einen großen Nachteil: Crawler und Spider (die für die Indizierung und Bewertung der Webseiten zuständig sind) können diese nicht oder nur sehr schwer lesen und daher auch nicht indizieren. Die Indizierung ist allerdings notwendig, um überhaupt auf einer Suchergebnisseite gelistet zu werden. Demzufolge ist es nachvollziehbar, dass Webseiten und Inhalte, die auf Flash basieren, nicht in das Ranking von Suchmaschinen mit aufgenommen werden. Die Suchmaschinenbetreiber suchen immer wieder nach einer Lösung zu diesem Problem. Die jüngsten Ergebnisse sind das Auslesen von Texten und Links aus Flash-Daten oder externen Quellen wie XML-Dateien, die die Inhalte für die Flash-Seiten beinhalten. Trotz allem ist die Indizierung bei Flash bei weitem nicht so effektiv wie bei HTML. Bei einer multimedialen Flash-Anwendung ist es zum Beispiel nicht möglich Bilder, Video- oder Audioinhalte und deren Bedeutung zu extrahieren und die Zuordnung von Unterseiten zur Navigation der Webseite ist nur äußerst umständlich über das sogenannte Deeplinking möglich. Wobei dieses Problem auch bei anderen Webtechnologien, wie zum Beispiel AJAX, mit dessen Hilfe Inhalte in die aktuell geöffnete Webseite nachgeladen werden können, bestehen.

Bei der Verwendung von Adobe Flash und der Suchmaschinenoptimierung bietet Google einige Ratschläge, die helfen sollen: Suchmaschinen-Robotern sollen stehts Zugriff auf die selben Inhalte wie die Nutzer haben. Das bedeutet, dass keine zusätzlichen Inhalte aussschließlich für Crawler und Spider zur Verfügung gestellt werden sollen. Die Angabe von Informationen die nur von Suchmaschinen-Robotern gelesen werden können nennt man Cloaking. Grob umschrieben gibt man den Robotern dabei vor etwas zu sein was man nicht ist. Weitere Empfehlungen von Google sind, auf die Erstellung von vollständig auf Flash basierenden Webseiten zu verzichten und stattdessen gezielt einzelne Flash-Inhalte, zum Beispiel Videos, in eine HTML Seite zu integrieren. Eine Alternative dazu wäre die gewünschten Inhalte sowohl mit HTML5 als auch mit Adobe Flash zu erstellen, womit die Roboter die HTML-Version indizieren können und die Nutzer die eventuell besser aufbereitete Flash-Version besuchen können.

Das ein Suchmaschinenbetreiber seinen Nutzern Ratschläge bei der Erstellung von Flash-Inhalten bereitstellt, zeigt wie umständlich es sein kann, diese für Suchmaschinen zu optimieren. Für eine möglichst erfolgreiche Optimierung der Suchergebnisse sollte stets ein externes Datenformat wie HTML oder XML eingesetzt werden. Damit ist es wesentlich einfacher, ein optimales Suchmaschinenergebnis zu erhalten. Auch multimediale Inhalte können durch einfache Mittel

mit Bedeutung versehen werden, die wiederum für die Crawler und Spider einfach verständlich sind. Unter anderem sind die weiteren Vorteile bei der Nutzung von HTML die Möglichkeit, Hyperlinks unterschiedlich stark zu bewerten, Meta-Attribute wie Stichwörter und/oder eine Beschreibung der Seite anzugeben und damit eine differenziertere Indizierung zu erreichen und somit ein deutlich besseres Ergebnis bei Suchmaschinen zu erreichen.

Ist es für den Betreiber einer Webseite wichtig, im Ranking von Suchmaschinen weit vorne zu liegen, führt fast kein Weg an HTML vorbei. Adobe Flash Inhalte können, wie sich gezeigt hat, auch indiziert werden, allerdings sind deren Relevanz für Suchmaschinen nie so hoch wie die von HTML Inhalten. Mit der neuen Version von HTML stehen nun noch mehr Möglichkeiten zum Einbinden von multimedialen und interaktiven Inhalten zur Verfügung, die wiederum eine optimale Alternative für die Optimierung von Suchmaschinenergebnissen genutzt werden können.

6.5.3. Schnittstellen

HTML5 und Adobe Flash sind mächtige Entwicklertools, welche die Realisierung komplexer Anwendungen ermöglichen. Dennoch ist es mittlerweile im Internet notwendig, dass Schnittstellen zu anderen Technologien entweder bereits vorhanden oder zumindest realisierbar sind. Technologien, die oft in einem Verbund verwendet werden, sind Datenbanken (SQL), Auszeichnungs- und Austauschsprachen wie XML oder JSON oder Programmiersprachen wie PHP oder JAVA.

Da HTML5 eine Auszeichnungssprache und keine Programmiersprache ist, können Inhalte und Daten aus anderen Technologien nur dargestellt werden. Mit Hilfe von JavaScript lassen sich diese Daten allerdings auch verarbeiten. Demnach ist im Fall von HTML5 nicht die Frage, ob die Technologie ausreichende Schnittstellen bietet, sondern ob es genügend Möglichkeiten zur Integration verschiedener andersartiger Technologien bietet und ob diese Technologien auf JavaScript-Aufrufe reagieren können. Daher müssen die Daten der eigesetzten Technologie so aufbereitet werden, dass sie mittels HTML dargestellt werden können, zum Beispiel als Text, Bild oder sonstigen multimedialen Inhalt.

Adobe Flash bietet ebenfalls Möglichkeiten für den Austausch von Daten zwischen verschiedenen Technologien an. Allerdings müssen, wie bei HTML5, sämtliche Daten im Voraus aufbereitet werden um innerhalb von Flash verarbeitet und dargestellt werden zu können. Mit der, in Flash integrierten Programmiersprache ActionScript, lassen sich die Daten in vielfältigerer Weise weiterverarbeiten, als es mit purem HTML5 möglich wäre. Dennoch kann mit Flash nicht direkt auf andere Technologien zugegriffen werden.

7. Implementierungen

Um die in dieser Arbeit vorgestellten Technologien im praktischen Einsatz zu veranschaulichen beschreibt dieses Kapitel die Implementierungen von drei unterschiedlichen Applikationen. Diese Applikationen verwenden jeweils verschiedene Features der HTML5 Spezifikation, wobei die Aufmerksamkeit hauptsächlich an der Nutzung des Canvas Elements liegt, und Adobe Flash mit ActionScript 3.0.

7.1. Animation

7.1.1. Konzept

7.1.2. Umsetzung HTML5

7.1.3. Umsetzung Adobe Flash

7.2. Spiel

7.2.1. Konzept

7.2.2. Umsetzung HTML5

Dokumente und Objekte

Bei der Umsetzung dieses Spiels werden HTML, CSS und JavaScript-Dateien benötigt. Zusätzlich werden Grafiken benötigt, die die Juwele im Spiel repräsentieren soll.

1. Beginn Index.html - Benötigte CSS+JS Files festlegen. settings festlegen yepnope preload/loader drawImage drawRect clip save/restore Rotation/Translate/Scale

7.2.3. Umsetzung Adobe Flash

7.3. Multimedia-Anwendung

7.3.1. Konzept

7.3.2. Umsetzung HTML5

7.3.3. Umsetzung Adobe Flash

7.4. HTML5 Frameworks for Animations, Games and more

- kineticjs (http://www.kineticjs.com) f
- fabricjs (http://fabricjs.com/) f
- paperjs (http://paperjs.org/) f

- createjs (http://www.createjs.com) f
- pixijs (http://www.pixijs.com/) f
- processingjs (http://processingjs.org/) f
- impactjs (http://impactjs.com/) k
- limejs (http://www.limejs.com/) f
- craftyjs (http://craftyjs.com/) f
- canvasengine (http://canvasengine.net/) f
- threejs (http://threejs.org/) f
- bhive (http://www.bhivecanvas.com/) f
- cakejs (https://code.google.com/p/cakejs/) f
- phaser.io (http://phaser.io/) f
- twojs(http://jonobr1.github.io/two.js/) f

8. Zukunftsaussicht

8.1. Wird HTML5 Flash ersetzen?

Steve Jobs deutet in seinem offenen Brief immer wieder an, dass Flash veraltet und HTML5 die Zunkunft ist. Dies ist insofern falsch, da HTML5 mit seiner bisherigen Browserunterstützung noch zu weit entfernt ist, um mit dem Flash Player mitzuhalten. Unrecht hat Steve Jobs mit der Aussage der fehlenden Offenheit allerdings nicht. Flash soll eingesetzt werden, wenn das Ziel mit einem offenen Standard nicht erreicht werden kann. Mit Flash kann qualitativ hochwertige Leistung erbracht werden. Aufgrund seiner Verbreitung ist es ein inoffizieller Standard.

HTML5 steckt noch in den Kinderschuhen und hat es schwer, mit Flash zu konkurrieren. Denn auch wenn Flash geschlossen ist, ist die Verbreitung des Flashplayer immer noch höher als die Verbreitung von HTML5-fähigen Browsern. Außerdem hat es ein Unternehmen wie Adobe wesentlich einfacher, ein neues Feature in ihren Player einzuführen, als Konsortien wie die W3C oder WHATWG, welche jedes Feature auf ihre Art demokratisch einführen und dann auf Browsersupport hoffen.

Im Videbereich kann sich HTML5 zwar schon präsentieren, doch auch auf YouTube wird der Flash Player weiterhin eine wichtige Rolle spielen, da es noch an wichtigen Funktionen in HTML5 Video fehlt, wie z.B. an einem nativen Vollbild, einem Schutz vor Download zur Wahrung der Urheberrechte oder der Kommunikation mit dem Nutzer. YouTube bietet dem Nutzer die Möglichkeit, direkt über die Webcam ein Video hochzuladen, was ohne Flash nicht möglich wäre.

Entscheidend dafür, was demnächst in HTML5 und was weiterhin in Flash umgesetzt wird, werden sicher die Interessen der Webentwickler sein. Während der Anwender nämlich nur die Technik nutzt und es ihm egak sein wird wie sein Video oder seine Webseite läuft, liegt es an dem Entwickler, ob er überhaupt in relativ funktionsarmen Javascript statt des sehr umfangreichen ActionScript 3 programmiert. Große Unternehmen wie Yahoo, Facebook oder Google könnten ihre Vorteile wiederum aus Webstandards ziehen, da sie selbst eigene Anforderungen an eine neue Spezifikation einbringen können, ohne dass sie von einem kommerziellen Anbieter wie Adobe abhängig sind.

Die Frage, ob HTML5 den Flash Player in Zukunft ablösen wird, bleibt also offen. Flash wird uns sicherlich noch eine Weile erhalten bleiben, da mit HTML5 in seiner jetztigen Form und Verbreitung noch nicht diesselbe Masse erreicht werden kann. Sicher ist auch, dass keines der beiden Programme das andere vollständig ersetzen kann, weshalb man auch nicht von Konkurrenz sprechen sollte, sondern von gegenseitiger Ergänzung.

So eignet sich Flash in Zukunft vo allem für Elemente, die eine hohe Performance benötigen und auf eine umfrangreiche Scriptbibliothek (ActionScript 3) zugreifen können, wie z.B. Spiele, Rich Internet Applications, komplexe 3D-Animationen, Audio/Videoplayer, Simulationen oder umfangreiche Präsentationen mit hohem Audio- und Video-Anteil. Die HTML5/JavaScript

Funktionalitäten eigenen sich hingegen für interaktive Webseitenelemente (Accordions, Tooltips, Dropdown-Menüs, Tabs, uvm), Formularvalidierung, Chats oder einfache Präsentationen.

8.2. HTML6

Die Unterstützung von HTML5 ist noch nicht gewährleistet - hier und da wird an allen Ecken noch gebastelt. Aber trotzdem hat im Januar der Google-Mitarbeiter und das WHATWG-Mitglied Mark Pilgrim im WHATWG-Blog bereits einen Einblick in HTML6 gewährt. Unter anderem wurde ein neues Element mit dem Namen *<device> vorgestellt, welches z.B. Webcam-Konferenzen ohne den Umweg über Flash ermöglichen würde. Aber ob diese Spezifikation auch wirklich „HTML6" heißen wird, ist unklar. Weiter schreibt Pilgrim nämlich:*

> *The next version of HTML doesn't have a name yet. In fact, it may never have a name, because the working group is switching to an unversioned development model. Various parts of the specification will be at varying degrees of stability, as noted in each section. But if all goes according to plan, there will never be One Big Cutoff that is frozen in time and dubbed „HTML6". HTML is an unbroken line stretching back almost two decades, and version numbers are a vestige of an older development model for standards that never really matched reality very well anyway. HTML5 is so last week. Let's talk about what's next.*

Pilgrim sagt also, dass die Vergabe von Versionsnummern veraltet ist und besonders für den Veröffentlichungsprozess von HTML-Standards nicht funktioniert. Ebenfalls interessant ist, dass der HTML5 WHATWG-Entwurf seit diesem Eintrag seinen Namen immer wieder von „WHATWG HTML (including HTML5)" zu „HTML5 (including next generation addition still in development)" ändert.

All dies könnte darauf hindeuten, dass HTML5 wirklich die letzte versionierte HTML-Spezifikation ist. Die Zukunft heißt also nicht „HTML6", sondern einfach nur „HTML" oder „HTML5 mit zusätzlichen Elementen". Das würde die Entwicklung vorantreiben. Es muss nicht mehr ein ganzes Paket an Neuerungen herausgebracht werden, auch einzelne Elemente können eingeführt werden. Schließlich sind es die Browserhersteller, die im Endeffekt HTML5 entwerfen. Dieses könnte eventuell die Existenz der W3C in Zukunft überfküssig machen. Die beiden Spezifikationen sind beinahe identisch und alles, was sich in der W3C-Version findet, steht auch in der WHATWG-Version. Iam Hickson hat allerdings in einer E-Mail kurz erwähnt, dass geplant ist, die WHATWG-Spezifikation in Zukunft ohne engere Absprachen mit dem W3C zu erweitern. Das neue <device> Element ist also nur ein Anfang.

> I've given up trying to keep a WHATWG copy of the HTML5 spec that matches what the W3C publish [...]

Auch Buchautor Peter Kröner prophezeit, dass es eventuell nicht einmal zu einer W3C-Version von HTML5 kommt.

> Die spannende Frage ist, ob es den „One Big Cutoff", sprich den Recommendation-Status für HTML5 zu einem Zeitpunkt, an dem es noch jemanden interessiert, zumindest für HTML5 geben wird. Ich glaube nicht.

Dies begründet Kröner damit, dass die Ausarbeitung von HTML5 seit dem Beginn der Entwicklung im Jahr 2004 bis heute (2013) neun Jahre gedauert hat und inzwischen zum großen Teil implementiert und benutzbar ist. Trotzdem wird aber von einem Recommendation-Status gesprochen, der erst im Jahr 2022 angesetzt wird. Wenn im Jahr 2013 nun aber bereits von einem HTML6 *<device> Element geredet wird, könnte bereits im Jahr 2016 ein Standard namens HTML6 entwickelt und großteilig in den Browsern implementiert sein und infolgedessen den Stand von HTML5 im Jahr 2013 haben. Also ein Datum, was neun Jahre vor dem Recommendation-Status von HTML5 liegt und im Jahr 2022 wahrscheinlich niemanden mehr interessieren wird. Auch aus dieser Sicht scheint ein versionsloses HTML-Modell mehr Sinn zu machen und ist laut Kröner bereits auf dem Weg:*

> *Wenn die WHATWG obendrein plant, mit HTML6 ein neues, versionsloses Entwicklungsmodell einzuführen, betrifft dies bereits HTML5. Die Spezifikationen der WHATWG sind schon jetzt ein Mischmasch aus alten HTML-Features, Neuerungen, die sich auf einen breiten Konsens stützen hoch kontroversen Ideen wie HTML5-Microformats und dem* <device> Element. Der unversionierte Ausbau ist also bereits im Gange."

Literaturverzeichnis

[1] *D. Geary*, Core HTML5 Canvas - Graphics, Animation and Game Development, *1st ed. Prentice Hall, May 2012.*

[2] *S. Fulton and J. Fulton*, HTML5 Canvas, *1st ed. O'Reilly Media, May 2011.*

[3] *R. Hawkes*, Foundation HTML5 Canvas: For Games and Entertainment, *1st ed. friendsofED, April 2011.*

[4] *B. Lamberta and K. Peters*, Foundation HTML5 Animation with JavaScript, *1st ed. friendsofED, November 2011.*

[5] *Makzan*, HTML5 Games Development by Example: Beginner Guide, *1st ed. Packt Publishing, August 2011.*

[6] *S. Koch*, JavaScript: Einführung, Programmierung und Referenz, *6th ed. Dpunkt Verlag, August 2011.*

[7] *T. Hauser, A. Kappler, and C. Wenz*, Das Praxisbuch ActionScript 3: Aktuell zu Adobe Flash CS5, *1st ed. Galileo Design, September 2010.*

[8] *D. Winnie*, Fundamentals of ActionScript 3.0: Develop and Design, *1st ed. Peachpit Press, Juli 2011.*

[9] *E. Feronato*, Flash Game Development by Example, *1st ed. Packt Publishing, Mar 2011.*

[10] *K. Peters*, Foundation ActionScript 3.0 Animation: Making Things Move!, *1st ed. friendsofED, April 2007.*

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

3G	*3rd Generation Mobile Telecommunications*
Ajax	*Asynchronous JavaScript and XML*
AS3	*ActionScript3*
API	*Application Programming Interface (Programmierschnittstelle)*
CSS	*Cascading Style Sheet*
CSS3	*Cascading Style Sheet Level 3*
DOM	*Document Object Model*
HTML	*HyperText Markup Language*
HTML5	*HyperText Markup Language Version 5*
JS	*JavaScript*
JSON	*JavaScript Object Notation*
RIA	*Rich Media Application*
SGML	*Standard Generalized Markup Language*
SQL	*Structured Query Language*
SVG	*Scaleable Vector Graphics*
UA	*User Agent*
URL	*Uniform Resource Locator*
W3C	*World Wide Web Consortium*

A. Überschrift des ersten Anhangs

Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text ...

B. Überschrift des zweiten Anhangs

Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text …