André Langer

Evaluierung von AJAX-basierten Frameworks für das Web 2.0

Studienarbeit

GRIN

BEI GRIN MACHT SICH IHR
WISSEN BEZAHLT

- Wir veröffentlichen Ihre Hausarbeit,
  Bachelor- und Masterarbeit

- Ihr eigenes eBook und Buch -
  weltweit in allen wichtigen Shops

- Verdienen Sie an jedem Verkauf

Jetzt bei www.GRIN.com hochladen
und kostenlos publizieren

GRIN

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de/ abrufbar.

Dieses Werk sowie alle darin enthaltenen einzelnen Beiträge und Abbildungen sind urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsschutz zugelassen ist, bedarf der vorherigen Zustimmung des Verlages. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen, Auswertungen durch Datenbanken und für die Einspeicherung und Verarbeitung in elektronische Systeme. Alle Rechte, auch die des auszugsweisen Nachdrucks, der fotomechanischen Wiedergabe (einschließlich Mikrokopie) sowie der Auswertung durch Datenbanken oder ähnliche Einrichtungen, vorbehalten.

Impressum:

Copyright © 2007 GRIN Verlag
ISBN: 9783638632805

Dieses Buch bei GRIN:

https://www.grin.com/document/74060

André Langer

Evaluierung von AJAX-basierten Frameworks für das Web 2.0

GRIN Verlag

GRIN - Your knowledge has value

Der GRIN Verlag publiziert seit 1998 wissenschaftliche Arbeiten von Studenten, Hochschullehrern und anderen Akademikern als eBook und gedrucktes Buch. Die Verlagswebsite www.grin.com ist die ideale Plattform zur Veröffentlichung von Hausarbeiten, Abschlussarbeiten, wissenschaftlichen Aufsätzen, Dissertationen und Fachbüchern.

Besuchen Sie uns im Internet:

http://www.grin.com/

http://www.facebook.com/grincom

http://www.twitter.com/grin_com

TECHNISCHE UNIVERSITÄT
CHEMNITZ

Studienarbeit

Evaluierung von AJAX-basierten frameworks für das Web 2.0

Vorgelegt von André Langer

30. März 2007

Fakultät für Informatik
Professur Verteilte und Selbstorganisierende Rechnersysteme

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides Statt, dass ich die vorliegende Arbeit selbstständig angefertigt, nicht anderweitig zu Prüfungszwecken vorgelegt und keine anderen als die angegebenen Hilfsmittel verwendet habe. Sämtliche wissentlich verwendete Textausschnitte, Zitate oder Inhalte anderer Verfasser wurden ausdrücklich als solche gekennzeichnet.

Chemnitz, den 30. März 2007

Zusammenfassung

„Remote Scripting“-Anwendungen erleben seit einigen Jahren einen regelrechten Anfrageboom. Während aus usability-Sicht bisher eine strikte Unterscheidung zwischen Desktop-Anwendungen und Webapplikationen herrschte, finden sich seit einiger Zeit zunehmend Angebote im World Wide Web, die diese strikte Trennung verwischen lassen. Interaktive Nutzerdialoge, nebenläufige Prozessabarbeitung und visuelle Unterstützungsmittel wie Drag & Drop- Effekte halten auf Webseiten Einzug, die dem Nutzer bisher nur aus eigenständigen Softwareprodukten in einer spezifischen Betriebssystemumgebung bekannt waren. Viele dieser neuen Anwendungs- und Interaktionsmöglichkeiten im weltweiten Datennetz werden inzwischen unter dem Oberbegriff Web 2.0 zusammengefasst. Für den Nutzer bringt dieser neue Entwicklungstrend viele Vorteile: Ansprechende, intuitive Nutzerführungen ohne die Notwendigkeit, eine ganze Internetseite bei jedem Interaktionsschritt neu zu laden und ohne bemerkbaren zeitlichen Overhead.

Was für den Nutzer Erleichterung bringen soll, bedeutet häufig für einen Programmierer zunächst Mehraufwand. Eine Technik zur Realisierung solcher so genannten Rich Internet Applications, die sich in den letzten beiden Jahren immer mehr in den Vordergrund gedrängt hat, wird unter der Bezeichnung AJAX zusammengefasst. Einen einheitlichen Standard gibt es dabei nicht, sodass fast täglich neue AJAX-basierte frameworks veröffentlicht werden, die dem Programmierer (wenigstens einen Teil der) Komplexität der Programmflusssteuerung abnehmen sollen. Aufgabe der Studienarbeit soll es daher sein, das inzwischen unüberschaubar gewordene Angebot an AJAX frameworks zu systematisieren und einen Überblick über Vor- und Nachteile ausgewählter Programmbibliotheken zu geben. Dafür ist ein Kriterienkatalog zu erarbeiten, der eine Bewertung der verschiedenen frameworks nach unterschiedlichen Gesichtspunkten ermöglicht. Besonderer Schwerpunkt ist dabei auf Kriterien aus Programmierersicht (Sprachunabhängigkeit, Overhead, Implementierungsmöglichkeiten,…) und Anwendersicht (Plattformanforderungen, Einarbeitungszeit, Ergebnisqualität, …) zu legen. Auf den Kriterienkatalog ist anschließend eine Auswahl an bereits existierenden, frei verfügbaren AJAX frameworks anzuwenden, die als zukünftig relevant eingeschätzt werden. Die Ergebnisse sind abschließend in einer Gesamtübersicht zu präsentieren, die eine objektive Empfehlung für Nutzer darstellen soll, die vor der Wahl stehen, welche AJAX Programmbibliothek sie zukünftig einsetzen sollten.

Inhaltsverzeichnis

ZUSAMMENFASSUNG III

INHALTSVERZEICHNIS IV

ABBILDUNGSVERZEICHNIS VII

TABELLENVERZEICHNIS VIII

LISTINGS IX

ABKÜRZUNGSVERZEICHNIS X

1. EINLEITUNG 1

1.1. AJAX – eine kurze Einführung 1

1.2. Zielsetzung der Arbeit 5

1.3. Aktueller Stand 6

2. GRUNDLEGENDE BETRACHTUNGEN 9

2.1. Geschichtlicher Ursprung von AJAX 9

2.2. Begriffsklärung 13

2.2.1. MVC 13

2.2.2. Remote Scripting 14

2.2.3. RIA 14

2.2.4. Widget 15

2.2.5. Wrapper 15

2.2.6. Stub 15

2.3. Definition der Anwendungsdomäne 16

2.3.1. Web Remoting 16

2.3.2. DOM-Manipulation 18

2.3.3. Widgets 18

2.3.4. Visuelle Effekte 19

2.3.5. Browseranwendungen 19

2.4. Beispiel-Szenarien 21

2.4.1. „Hello World“ example 21

2.4.2. Adresskartei 25

2.4.3. AJAX-Bildergalerie 30

2.5. Was ist AJAX? 33

3. AJAX FRAMEWORKS 37

3.1. Der Begriff „framework“ 37

3.2. Abgrenzung zu Funktionsbibliotheken 38

3.3. Anforderungen an ein framework 39

3.4. Klassifikation 41

3.5. Überblick ... **43**
4. BESCHREIBUNG DER EVALUIERUNG ... **47**
4.1. Allgemeiner Überblick ... **47**
4.2. Testauswahl ... **47**
4.3. Beschreibung der Durchführung ... **48**
4.4. Testumgebung ... **49**
4.5. Kriterien ... **50**
4.6. Bewertungsmaßstab ... **53**
5. DURCHFÜHRUNG DER EVALUIERUNG ... **55**
5.1. Clientframeworks ... **55**
5.1.1. Javascript-basierte Bibliotheken (Basisframeworks) ... 55
5.1.1.1. ACE ... 55
5.1.1.2. AjaxToolbox ... 56
5.1.1.3. Bajax ... 57
5.1.1.4. HTMLHttpRequest ... 58
5.1.1.5. Lokris ... 59
5.1.1.6. MAJAX ... 60
5.1.1.7. Prototype ... 61
5.1.2. Javascript-basierte Effektbibliotheken (Applicationframeworks) ... 63
5.1.2.1. Adobe Spry ... 63
5.1.2.2. DOJO ... 64
5.1.2.3. jQuery ... 65
5.1.2.4. MochiKit ... 66
5.1.2.5. Mootools ... 67
5.1.2.6. Yahoo User Interface library ... 67
5.1.3. Basis- und Applikationsframeworks erweiternde frameworks ... 70
5.1.3.1. Freja ... 70
5.1.3.2. OpenRico ... 71
5.1.3.3. Script.aculo.us ... 73
5.2. Serverframeworks ... **74**
5.2.1. PHP ... 74
5.2.1.1. AJAXAgent ... 74
5.2.1.2. Flexible AJAX ... 75
5.2.1.3. My-BIC ... 76
5.2.1.4. Sajax ... 78
5.2.1.5. tinyAjax ... 79
5.2.1.6. XAJAX ... 80
5.2.2. Perl ... 82
5.2.2.1. Catalyst ... 82

5.2.2.2. CGI::AJAX ... 83
5.2.3. Python ... 85
5.2.3.1. CherryPy ... 85
5.2.3.2. Nevow ... 86
5.2.4. Java ... 88
5.2.4.1. DWR ... 88
5.2.4.2. Google Web Toolkit ... 89
5.2.4.3. JSON-RPC-Java ... 92
5.2.5. DotNet ... 94
5.2.5.1. AJAX.NET ... 94
5.2.5.2. Anthem.Net ... 95
5.2.5.3. ASP.NET AJAX (Codename 'Atlas') ... 96
5.2.5.4. ComfortASP.NET ... 99
5.2.5.5. Visual WebGUI ... 100
6. RESULTATE ... 103
6.1. Übersicht ... 103
6.2. Fehlerbetrachtung ... 104
6.3. Testergebnisse ... 106
6.4. Bewertung der frameworks ... 110
6.4.1. Basisframeworks ... 110
6.4.2. Applicationframeworks ... 112
6.4.3. Basis- oder Applicationframeworks erweiternde Frameworks ... 114
6.4.4. Serverframeworks PHP ... 115
6.4.5. Serverframeworks Perl ... 117
6.4.6. Serverframeworks Python ... 117
6.4.7. Serverframeworks Java ... 119
6.4.8. Serverframeworks DotNet ... 120
6.5. Diskussion ... 121
7. AUSBLICK ... 125
7.1. Frameworkentwicklungen ... 125
7.2. Die Zukunft von AJAX ... 128
LITERATURVERZEICHNIS ... 131
INDEX ... 135
A. ANHANG ... A-3
A.1. Grafische Darstellung der Messwerte ... A-3
A.2. Übersicht über AJAX frameworks ... A-7
A.3. Testbögen ... A-15

Abbildungsverzeichnis

Abbildung 1: Standardbeispiel für neue Trends im Web 2.0: Google Suggest – Das Auto Completing ist in AJAX frameworks mit wenigen Anweisungen realisierbar 2

Abbildung 2: AJAX als "buzzword" in der "tag cloud" des Web 2.0 (Quelle: http://www.openjacob.org)..... 7

Abbildung 3: traditioneller Ablauf bei der Anforderung von Daten von einem Webserver: Die gesamte Website wird neu geladen, im Beispiel: eine Bilderauswahl unter http://www.flickr.com 16

Abbildung 4: Die gleiche Anfrage wie in Abb. 2 mittels AJAX, während die AJAX Engine auf das Eintreffen der HTTP Response wartet, kann der Nutzer die ursprüngliche Seite uneingeschränkt weiter benutzen 17

Abbildung 5: Mit AJAX verschwinden allmählich die Grenzen zwischen Desktop- und Webanwendungen, Bsp.: http://www.bindows.net 19

Abbildung 6: Schematische Einordnung von AJAX frameworks 43

Abbildung 7: Grafische Darstellung der Anzahl der Suchergebnisse von www.google.de bei der Suche nach der Kombination +ajax +{frameworkname}, Durchführung: 31.01.2007..... 45

Abbildung 8: Bewertung der getesteten frameworks im Schulnotensystem 107

Abbildung 9: Traffic bei Erstaufruf in kB in Boxplot-Darstellung 109

Abbildung 10: Ladezeit je Anwendungsszenario in s in Boxplot-Darstellung..... 109

Abbildung 11: Ladezeitvergleich basierend auf den einzelnen Anwendungsszenarien A3

Abbildung 12: Zeit zum Ausführen eines asynchronen Requests je framework in s A-4

Abbildung 13: übertragene Datenmenge bei Seitenerstaufruf je framework je Anwendungsbeispiel in kB. A-5

Abbildung 14: Messwertübersicht für die Abbildungen 11-13 ..... A-6

Tabellenverzeichnis

Tabelle 1: Bestandteile von AJAX 33

Tabelle 2: Einordnung von AJAX frameworks nach Kategorien 42

Tabelle 3: Klassifikation im Internet verfügbarer frameworks nach Sprachplattform 44

Tabelle 4: Übersicht über Anzahl der getesteten frameworks 48

Tabelle 5: Kriterienkatalog für Evaluierung 53

Tabelle 6: Meistgenutzte AJAX Toolkits im Vergleich (Quelle: [Ore06]) 126

Tabelle 7: Bedeutung wesentlicher IT-Technologien und der Zeitraum bis deren voraussichtliche Durchsetzung (Quelle: Computerzeitung/Gartner) 128

Listings

Listing 1: "Hello World" - klassische Realisierung mit PHP 21

Listing 2: Das "Hello World" Beispiel als AJAX-Realisation 22

Listing 3: Das "Hello World"-Beispiel in einer verbesserten AJAX-Variante 24

Listing 4: Eine mögliche Implementierung des Beispiels 2 mittels PHP 26

Listing 5: Das Adresskarteibeispiel basierend auf AJAX 29

Listing 6: Beispiel 3 in einer klassischen Realisierung mit PHP 31

Listing 7: Eine einfache, AJAX-basierte Bildergalerie 32

Listing 8: Beispiel 1 aus Kapitel 2.3.1 mit der ahah.js 36

Abkürzungsverzeichnis

AHAH	Asynchronous HTML and HTTP
AJAJ	Asynchronous JavaScript and JSON
AJAX	Asynchronous Javascript And XML
API	Application Programming Interface
ASP	Active Server Pages
CGI	Common Gateway Interface
CMS	Content Management System
CSS	Cascading Style Sheets
DHTML	Dynamic Hypertext Markup Language
DnD	Drag and Drop
DOM	Document Object Model
ECMA	Computer Manufacturers Association
FW	Framework
GUI	Graphical User Interface
ID	Identificator
IDE	Integrated Development Environment
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
JS	Javascript
JSF	Java Server Faces
JSNI	JavaScript Native Interface
JSON	JavaScript Object Notation
JSP	Java Server Pages
LOC	Lines of Code
MS	Microsoft
MVC	Model-View-Controller
OWA	Outlook Web Access
PHP	Hypertext Preprocessor
RIA	Rich Internet Application
RPC	Remote Procedure Call
SGML	Standard Generalized Markup Language
SQL	Structured Query Language
URL	Uniform Resource Locator
VS	Visual Studio
WWW	World Wide Web

XHR	XMLHttpRequest
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language
XSL	eXtensible Stylesheet Language
XSLT	eXtensible Stylesheet Language Transformation

1. Einleitung

1.1. AJAX – eine kurze Einführung

Bereits im antiken Griechenland verband man mit dem Namen AJAX vor allem eins: Macht, Schönheit und Größe. Die Rede ist hier von Ajax dem Großen, Sohn des Königs von Salamis, der in Homers Illiade in der Schlacht um Troja als Mutigster aller Griechen im Zweikampf dem Trojaner Hektor gegenüber stand. Der Kampf dauerte einen ganzen Tag, doch keiner der beiden Helden konnte ihn für sich entscheiden, weswegen beide letztendlich den Kampf mit Respekt gegenüber dem Anderen beendeten. [Fid07] Jahrhunderte lang wurde der Mythos weitergetragen, doch ausgerechnet im Jahr 2005 tauchte der Name AJAX in einem vollkommen neuen Zusammenhang wieder auf: Wie ein griechischer Held „eroberte“ der Begriff binnen kurzer Zeit die Welt des Internets und selbst zwei Jahre später kann noch niemand einen Orakelspruch sprechen, wohin der Weg von AJAX im „Web 2.0“ in nächster Zeit führen wird. [Hos07]

AJAX – dies ist zu Beginn des 21. Jahrhunderts eine Art Modewort („buzzword“), mit dessen Techniken es möglich sein soll, die Bedienung des klassischen World Wide Web zu revolutionieren[1] oder zumindest zu vereinfachen und das Nutzungskonzept einer Webseite stärker auf die Bedürfnisse der Nutzer auszulegen. Interessant dabei ist, dass sich hinter AJAX kein vollkommen neues Programmierkonzept oder sogar eine eigene neue Sprache verbirgt, sondern zunächst nur ein Akronym für „Asynchronous Javascript And XML“. Den sprichwörtlichen Stein ins Rollen gebracht hat dabei ein inzwischen vielfach zitierter Aufsatz von dem Amerikaner Jesse James Garrett unter dem Titel „Ajax: A New Approach to Web Applications“ im Februar 2005 [Gar05]. Seitdem hat sich einiges im Web getan: In den darauf folgenden Monaten entstanden Angebote im Internet, die Mitte der Neunziger Jahre noch kaum vorstellbar waren.[2] Diese Entwicklung setzte in einer Zeit ein, der mit der steigenden Verbreitung von so genannten Wikis und Blogs bereits ein weiterer Trend vorausgegangen war. Inhalte, die aktiv von Nutzern erstellt wurden, rückten immer mehr in den Mittelpunkt des Interesses – der Nutzer selbst verbunden mit möglichst einfach und intuitiv zu bedienenden Benutzeroberflächen wurde zunehmend wichtiger.

[1] Etwas vereinfacht betrachtet hat sich das zugrunde liegende Konzept des WWW seit 1989 mit der Einführung der Hyper-Text Markup Language am CERN nicht mehr wesentlich geändert.

[2] An dieser Stelle wird in Publikationen und Artikeln häufig auf bekannte Beispiele wie GoogleMaps (http://maps.google.com), GoogleSuggest (http://labs.google.com/suggest/) oder beispielsweise Flickr (http://www.flickr.com) verwiesen.

Die Techniken dafür gab es bereits vorher; so wurde zum Beispiel eine Vielzahl von Content Management Systemen entwickelt, mit der auch ein Nutzer ohne Programmierkenntnisse schnell Inhalte im Web publizieren konnte und die entsprechenden Seiten dynamisch im Hintergrund generiert und als HTML-Seiten ausgeliefert wurden. Doch im Gegensatz dazu wirkten Internetseiten, die im Hintergrund auf Techniken wie AJAX, XML und Webservices aufsetzten, plötzlich wesentlich lebendiger und „reicher“ an Funktionalitäten.

Abbildung 1: Standardbeispiel für neue Trends im Web 2.0: Google Suggest – Das Auto Completing ist in AJAX frameworks mit wenigen Anweisungen realisierbar

Im Herbst 2004 beschäftigte sich in San Francisco eine Konferenz unter dem Titel „Web 2.0“ mit dieser Entwicklung. Community-Orientierung, erhöhte Benutzerfreundlichkeit, eine intuitive Bedienung und das allmähliche Verwischen von Dienstgrenzen durch Kombination bestehender Techniken und Inhalte („Mashup“) wurden auf der Konferenz als wesentliche Aspekte identifiziert, sowie weitere Schlüsselprinzipien für die kommenden Jahre benannt. Dale Dougherty – ein Mitorganisator der Konferenz vom O’Reilly Verlag – stellte eine Reihe von Vergleichen an, in welchem Veränderungsprozess sich das Internet aktuell befinden würde, und welche altbekannten Services in Zukunft durch neuen Technologien ersetzt werden würden. Der Name der Konferenz steht seitdem als Schlagwort für diesen gesamten Veränderungsprozess. „Web 2.0 ist heute, Web 1.0 war gestern“ wird beispielsweise auf der Website von Helge Städtler unter dem Titel „Tachometer der Webentwicklung“ [Sta07] getitelt[3].

[3] Dass sich aufgrund derartiger Slogans und Schlagwörter schnell ein nur schwer zu kontrollierender Hype entwickeln kann, bei dem mit zunehmendem Fortschreiten Begrifflichkeiten und ursprüngliche Konzepte immer stärker verwässern, wird am Beispiel Web 2.0, AJAX und Mashup schnell deutlich. So verwundert es nicht, dass bereits im Jahr 2006 erste Spekulationen wie „Überholt das Web 3.0 gar das Web 2.0“ [Bas06] in einigen Weblogs zu finden waren, welche zumeist ohne wissenschaftlichen Hintergrund von Unternehmen und anderen selbst ernannten Experten zu Marketingzwecken genutzt wurden.

Rückblickend kann wohl festgestellt werden, dass AJAX als Programmierkonzept an dieser Entwicklung einen maßgeblichen Anteil hat und weiterhin haben wird. Die Grenzen zwischen Web-Applikationen und Desktop-Anwendungen scheinen allmählich zu verschwinden. Plötzlich entdecken Web-Designer wieder „Drag and Drop"-Effekte sowie weitere Animationsarten für sich, Websitebesuchern werden Hilfeassistenten angeboten und reine textuelle Informationen werden grafisch ansprechend aufbereitet und sind von Nutzerseite aus jederzeit anpassbar.

Zu erwähnen ist dabei, dass diese Sonderfunktionen nicht mehr Ressourcen benötigen als die althergebrachten Websiteinhalte der letzten Jahre. Eher im Gegenteil kann gesagt werden, dass auf AJAX-basierten Internetseiten ein völlig neues „look-and-feel" entsteht. Websiteinhalte scheinen spürbar schneller zu laden, jeder Mausklick ruft anscheinend eine unmittelbare Aktion hervor, und das gesamte Timingverhalten von Webanwendungen scheint sich unter Einsatz dieser neuen Technik zu verbessern, da ein wesentlicher Anteil aller Operationen nun auf Clientseite verlagert wird.

Der Mehraufwand für diese Reihe von Vorteilen ist zunächst auf Programmiererseite zu suchen, denn gegenüber traditionellen Programmieransätzen auf Serverseite (ASP, PHP, Perl, Python und andere) ist nun ein wesentlich komplexerer Programmablauf zu realisieren, der sowohl auf Server- als auch auf Clientseite stattfindet und ebenso eine wechselseitige Kommunikation zwischen diesen einschließt. Nicht nur die eindeutige Abarbeitung von Programmcode verwischt unter dem Konzept von AJAX, auch die Vielfalt der eingesetzten Sprachen stellt eine Herausforderung für die Entwickler von AJAX-basierten Webapplikationen dar. Während es vor Jahren genügte, Spezialkenntnisse in PHP oder Perl zu besitzen und das Layout der Internetseite einem Grafiker überlassen blieb, der mithilfe von Cascading Style Sheets die einzelnen Elemente einer Seite optisch ansprechend anordnete, verzahnen sich bei modernen Internetseiten diese Techniken immer mehr. Einzelne Skriptsprachen sind nicht mehr alleinstehend, sondern agieren zunehmend miteinander: So werden die Stärken von Javascript, XHTML / DOM, CSS, XML und Sprachen wie PHP, DotNet, Java sowie Datenbankanfragesprachen wie MySQL immer häufiger kombiniert, um noch ansprechendere Internetauftritte zu realisieren.

Diese Entwicklung mag aus Benutzersicht erfreulich sein, birgt auf Entwicklerseite aber mehrere Gefahren. Während seit Jahren stetig versucht wurde, Softwareentwicklung mithilfe von Standardisierungen, integrierten Softwareentwicklungsumgebungen (IDEs) und „Best Practise"-Empfehlungen in geregelte Bahnen zu lenken, besteht durch die Vermischung verschiedener Sprachen und Programmierkonzepte die Gefahr, die Bestrebungen ungewollt wieder umzukehren. Webseitenentwickler haben immer mehr Wahlmöglichkeiten, unter dem Einsatz welcher Techniken sie ein konkretes Problem lösen. Wird keine konsequente Umsetzung gefunden, entsteht schnell ein Mischcode, der schwer zu warten ist.

Eine weitere Gefahr ist, dass Webanwendungen aufgrund begrenzter Entwicklungsressourcen häufig nur für bestimmte Anwendungsfälle und Browserfamilien entwickelt werden. Auch wenn heutige moderne Browser größtenteils Webstandards mit dem gleichen Endverhalten umsetzen, so ist für einen Programmierer dennoch ein relativ hoher Aufwand mit vielen Fallunterscheidungen und Work-arounds nötig, um sowohl aktuelle Internetbrowser wie Internet Explorer, Mozilla/Firefox, Safari und Opera, als auch ältere Versionen dieser Programme zu unterstützen und alle Informationen einer Webseite identisch (oder zumindest in gleicher Art und Weise benutzbar) in diesen darzustellen.

Das skizzierte Problemszenario ist seit langem in der Softwareentwicklung bekannt und beschränkt sich nicht nur auf die Entwicklung von Webapplikationen. Um einen Programmierer bei seiner Arbeit zu unterstützen und ihm die Behandlung ständig wiederkehrender Problemszenarien zu vereinfachen, finden sich für nahezu jede Sprache eine Reihe von frameworks in Form von Programmbibliotheken. Dazu verbergen frameworks in unterschiedlichem Maße die zugrunde liegende Komplexität und abstrahieren von den technischen Realisierungen. Auch nach der rasanten Verbreitung von AJAX im Jahr 2005 wurden weltweit – ob nun von Privatpersonen oder größeren Unternehmen, ob kommerziell oder open-source - verschiedenste frameworks konzipiert, entwickelt und anderen Nutzern bereitgestellt. Knapp zwei Jahre später ist die Vielfalt dieser frameworks so groß geworden, dass es einem Einsteiger nur noch schwer möglich ist, einen Gesamtüberblick zu bekommen und herauszufinden, welche frameworks für welche Zwecke am Besten geeignet sind. Darüber hinaus ist es inzwischen schon nicht leicht, überhaupt zu unterscheiden, welche von den angebotenen AJAX-Programmbibliotheken wirklich AJAX-Komponenten mit erweiterten Funktionalitäten enthalten und welche sich unter Umständen nur so nennen, aber keinen Mehrwert bringen. Ebenso wenig ist häufig nicht genau erkennbar, inwieweit angebotene AJAX frameworks bereits einen stabilen Entwicklungsstand erreicht haben und ob sie auch in Zukunft unter der großen Konkurrenz weiter Bestand haben und unterstützt werden, oder ob in wenigen Monaten im schlechtesten Fall ein komplettes Reengineering des damit realisierten Projektes nötig werden könnte.

1.2. Zielsetzung der Arbeit

Ziel der vorliegenden Studienarbeit ist es, einen Überblick über bestehende AJAX frameworks zu geben und diese nach unterschiedlichen Kriterien zu bewerten. Datengrundlage dazu ist eine im Oktober 2006 durchgeführte Recherche, in der zunächst zum 31.10.2006 insgesamt 231 frameworks identifiziert wurden, die einen Bezug zu der Thematik „Asynchronous JavaScript and XML“ erkennen ließen. Schwerpunktmäßig wurden rein Javascript-basierte Funktionssammlungen, sowie frameworks basierend auf PHP, Perl, Python, DotNet und Java untersucht. Daneben existieren weitere frameworks für andere Sprachen, wobei hier exemplarisch Flash und Coldfusion genannt seien. Da Flash als Technologie bereits vor Jahren eine ähnliche Motivation wie AJAX verfolgte und die Thematik AJAX unter Flash eine eigene Studienarbeit füllen könnte, werden diese frameworks in der weiteren Betrachtung ausgeklammert. Das Gleiche gelte an dieser Stelle für Javascript-basierte frameworks mit einer reinen Spezialisierung auf Debugging und Logging, da diese aus Webseitenbesuchersicht ebenfalls keine Relevanz unter dem Schlagwort „Web 2.0“ aufweisen. Aus dem genannten Grund reduzierte sich die Zahl der aktuell verfügbaren framework um 22 auf insgesamt 209 Programmbibliotheken.

In einem zweiten Untersuchungsschritt konnte festgestellt werden, dass 18 der verbliebenen frameworks zwar überdurchschnittlich oft in Verbindung mit dem Begriff AJAX genannt wurden, selbst aber keine erkennbaren AJAX-ähnlichen Funktionalitäten implementierten. Die Projektwebsites von 6 weiteren frameworks, die auf mehreren anderen Internetseiten empfohlen wurden, waren sogar in mehreren Stichproben über einen Zeitraum von zwei Wochen generell nicht zu erreichen und ihr Entwicklungsstand erschien eingestellt. Insgesamt blieben deshalb als Datengrundlage für diese Studienarbeit 185 AJAX frameworks. Deren genaue Klassifikation wird im Kapitel 3.4 näher beschrieben. Eine Übersicht mit generellen Informationen zu allen AJAX frameworks ist im Anhang 1 zu finden. Die Übersicht erhebt aufgrund der derzeitigen sehr dynamischen Entwicklung in diesem Bereich dabei keinen Anspruch auf Vollständigkeit.

Die Studienarbeit besteht aus insgesamt sieben Teilen. Nach einer kurzen Einführung in die Thematik im vorliegenden Kapitel 1 beschäftigt sich Kapitel 2 zunächst mit einigen Grundlagen von AJAX, die für das spätere Verständnis der framework – Evaluation nötig sind. Die Ziele und Einsatzgebiete stehen dabei im Mittelpunkt und es wird versucht die Frage zu klären, was AJAX selbst eigentlich ist. Anhand von Fallbeispielen wird die Einsatzweise von AJAX geklärt, welche später als Grundlage für die Evaluierung der verschiedenen frameworks dienen. Kapitel 3 klärt darauf aufbauend zunächst grundlegende Begriffe im Zusammenhang mit AJAX frameworks und gibt einen Überblick über das breite Spektrum an AJAX frameworks.

In Kapitel 4 wird anschließend der Kriterienkatalog definiert, welcher der Durchführung der Evaluation und der Bewertung der AJAX frameworks zugrunde liegt und legt die Kriterien offen, unter welchen Gesichtspunkten die Auswahl an frameworks getroffen wurde, die von den knapp 200 Vertretern repräsentativ evaluiert wurden. Kapitel 5 stellt den experimentellen Teil der Studienarbeit dar, in dem kategorisiert nach den zugrunde liegenden Skriptsprachen einzelne frameworks untersucht und deren Vor- und Nachteile beschrieben werden. Daran anschließend gibt Kapitel 6 eine Übersicht über die gewonnenen Ergebnisse aus der Evaluierung und versucht, eine objektive Bewertung der getesteten frameworks abzugeben. Empfehlungen über gute und weniger gute frameworks für bestimmte Einsatzgebiete werden getroffen und deren Potential in Verbindung mit anderen Techniken diskutiert. Kapitel 7 schließlich versucht als Abschluss der Studienarbeit einen Ausblick auf die Zukunft von AJAX sowie AJAX frameworks anhand aktuell verfügbarer Fakten und Prognosen zu geben.

1.3. Aktueller Stand

Paradox an der Thematik „AJAX“ ist, dass sich hinter diesem Schlagwort im Grunde genommen keine neuen Technologien verbergen – Javascript, HTML und XML sind seit langem bekannt und wurden dementsprechend schon in einer Vielzahl wissenschaftlicher Arbeiten analysiert und vor wenigen Jahren auch standardisiert. Neu erscheint zwar zunächst der Ansatz der Asynchronität, aber auch dafür gab es die nötigen Techniken bereits 1998[4]. Warum erst zu Beginn des Jahres 2005 ein sprunghaftes Interesse an dieser Thematik aufkam, soll in Kapitel 2 geklärt werden. Die bedeutendste wissenschaftliche Arbeit zu dem Thema prägte Jesse James Garrett mit dem Titel „Ajax: A New Approach to Web Applications“ im Februar 2005 [Gar05]. Obwohl der Begriff „AJAX“ letztendlich ein Marketingbegriff für die Beschreibung von etwas bereits Bekanntem war, kam es in den Medien zu einem regelrechten Hype. Allein in dem Internet-Versandhaus amazon.com fanden sich im Februar 2006 knapp 200 verschiedene Buchtitel, die sich mit der Thematik AJAX beschäftigen, als Suchbegriff unter www.google.com waren es sogar geschätzte 92.900.000 Ergebnisse.[5] Zumeist beschäftigen sich diese Artikel mit einer Einführung in das Thema AJAX, vorwiegend für Programmierneulinge in diesem Gebiet, oder sind Gegenstand von threads in verschiedenen Diskussionsforen.

[4] Siehe Kapitel 2.1

[5] Gesucht wurde am 08.02.07 nach der Wortkombination „+ajax +xml“

Im Gegensatz dazu oder vielleicht gerade deswegen finden sich nur sehr wenige weitere wissenschaftliche Abhandlungen oder Veröffentlichungen, die sich mit der Thematik „Asynchronous Javascript“ oder sogar mit AJAX frameworks auseinandersetzen. Viele davon wurden erst innerhalb der letzten Monate im englischsprachigen Raum verfasst. Wissenschaftlich fundierte, deutschsprachige Artikel sind selten und finden sich zumeist als Veröffentlichung in speziellen Magazinen (entwickler.press). Exemplarisch sei hierfür das AJAXspecial vom Software & Support Verlag genannt.

Daneben erfolgt aktuell ein regelmäßiger Wissensaustausch auf verschiedensten Konferenzen zum Thema AJAX weltweit. Neben der Ursprungsveranstaltung „Web 2.0“ jedes Jahr im Oktober in San Francisco sei hier vor allem die „w-jax 2006“ in München und „Ajax in Action“ im Februar 2007 in Frankfurt genannt.

Obwohl Quellen im Internet wegen der kurzen Lebensdauer und der schwer einzuschätzenden Relevanz in der Regel kritisch zu sehen sind, sei abschließend an dieser Stelle auch die Seite www.ajaxpatterns.org erwähnt, die erstmals den Versuch startete, eine lückenlose Übersicht über bestehende AJAX frameworks zu geben.

Es finden sich eine Reihe weiterer derartiger Übersichten im weltweiten Datennetz, doch repräsentieren diese jeweils nur einen Bruchteil der tatsächlich angebotenen Programmbibliotheken oder beschäftigen sich tiefergehend mit einer bestimmten Art von AJAX frameworks wie zum Beispiel der ASP.Net-framework-Vergleich von Zeiss [Zei07] zeigt.

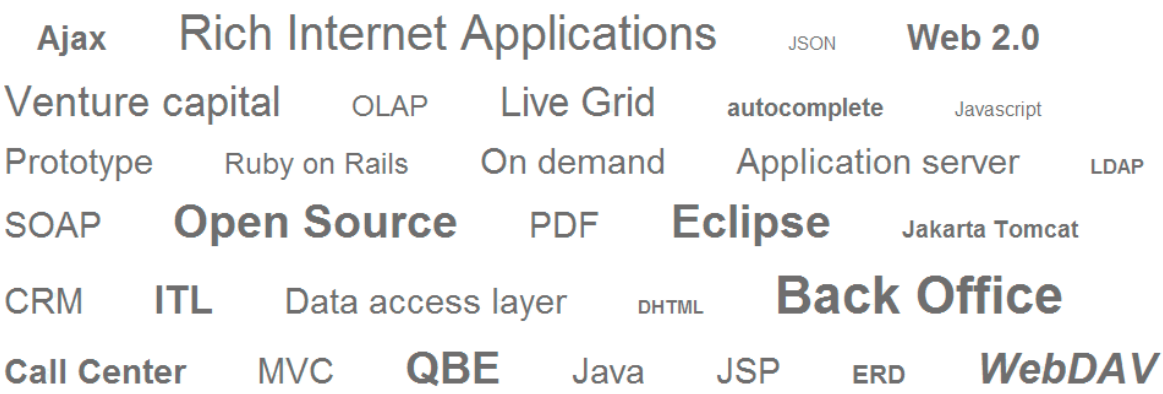

Abbildung 2: AJAX als "buzzword" in der "tag cloud" des Web 2.0 (Quelle: http://www.openjacob.org)