

Beispiele für den Einsatz von Ajax

Tunca Karabel
Hochschule für Technik

30. Januar 2007

Inhaltsverzeichnis

1. Einleitung	3
2. Einführung in Ajax Technik	4
2.1 Benutzte Sprachen und Methoden	6
2.2 Vergleich mit herkömmlichen Webanwendungen	6
2.3 Client-Server Plattform	8
3. Beispiele für den Einsatz	8
3.1 Anwendungen	8
3.2 Suchmaschinen	9
3.3 Webbrowser	10
3.4 Entwicklungsumgebungen	10
4. Probleme	11
4.1 Vorteile und Nachteile	11
4.2 Probleme und Lösungen	11
5. Zusammenfassung	12
Abkürzungsverzeichnis	13
Literaturverzeichnis	14

1 Einleitung

Jeden Tag surfen Millionen von Menschen im Internet und erhoffen sich ein schnelles Laden der Seiten, wenn sie ein Formular ausfüllen oder einfach eine Karte anschauen wollen. Auch wenn in erster Linie von der Geschwindigkeit oder Bandbreite des Internetzugangs abhängt, spielt Ajax dabei genauso eine große Rolle.

Der Begriff Ajax steht für „Asynchronous Javascript and XML*“ und wurde durch einen Artikel von Jesse James Garrett vom Februar 2005 geprägt. Dabei handelt es sich um den Einsatz verschiedener Technologien, die es möglich machen, Webseiten zu erstellen, die Arbeitsplatz-Anwendungen ähneln. Es ist im Grunde nichts anderes als das Zusammenspiel von HTML*, CSS*, DOM*, JavaScript und XML*. Diese sind Techniken, die bereits seit längerem existieren und die den meisten Webprogrammierern bekannt sind.

Die Wurzeln von Ajax stammen aber eigentlich von Zeiten des Internet Explorers 5 von Microsoft. Das Internet Explorer-Team integrierte damals eine Technologie in die Browser, um im Hintergrund HTTP*-Anfragen abzusetzen und die Rückgabe auszuwerten. Die Anforderung kam vom Office-Team, genauer gesagt von den Outlook-Entwicklern. Die Web-Schnittstelle von Outlook benötigte diese Funktionalität der HTTP-Anfragen im Hintergrund, um beispielsweise ohne permanentes Neuladen zu prüfen, ob es schon neue Mails gibt.

Andere bekannte Beispiele für den Einsatz von Ajax sind Web-Applikationen wie etwa Google Maps und MSN Virtual Earth, bei denen ohne sichtbares Neuladen der Seite Daten zwischen Browser und Server als XML abgefragt werden. Für den Benutzer ist das Ergebnis ähnlich wie bei normalen Arbeitsplatz-Applikationen. Wie diese Technik funktioniert und viele weitere Beispiele über Ajax werden im Weiteren in dieser Arbeit behandelt.

In dieser wissenschaftlichen Ausarbeitung wird die bekannte und häufig benutzte Internetseiten-Übertragungstechnik Ajax unter die Lupe genommen. Es werden Beispiele für den Einsatz gegeben, Vorteile und Nachteile aufgezählt und die Technik von Ajax näher beschrieben. Anhand von Beispielen und der Erläuterungen der Funktionsweise werden die Vorzüge und Nachteile von Ajax aufgezeigt.

* siehe Abkürzungsverzeichnis

2 Einführung in die Ajax Technik

Die Technik, die hinter Ajax steht, ist prinzipiell ganz einfach. Es ist inzwischen in jedem Browser als ein natives Objekt, genannt "XMLHttpRequest", implementiert. Das Prinzip von Ajax ist leicht zu verstehen. Der Hauptpunkt bei Ajax ist, die Server-Rückgabe auf der Client-Seite geeignet zu verarbeiten. Dazu werden JavaScript, in der Regel DOM*, aber auch manchmal DHTML* verwendet. Wie dieser Datenaustausch mit dem Server realisiert werden kann, wird mit einem kurzen Beispiel erklärt. Der Ablauf funktioniert in drei Schritten:

Als erstes wird das entsprechende HTTP-Element erzeugt. Beim Internet Explorer geht das folgendermaßen:

```
var http : new XMLHttpRequest("Microsoft.XMLHTTP");
```

Als zweites stellt man eine Verbindung zur Zielseite:

```
http.open("GET", "datei.html", true);
```

Der ‚true‘ am Ende steht dafür, dass die Skriptausführung weiter geht und nicht angehalten wird, bis die Daten vom Server zurückkommen. Sobald Daten vom Server zurückkommen, wird eine sogenannte *Callback-Funktion* aufgerufen.

```
http.onreadystatechange = ausgeben;
```

Am Ende muss man nur noch die Funktion definieren, mit der man die Rückgabe macht.

```
function ausgeben() {  
    if (http.readyState == 4) {  
        document.getElementById("Ausgabe").innerHTML = http.responseText;  
    }  
}
```

readyState ist 4, wenn das Objekt komplett geladen ist. 0 steht für nicht initialisiert, 1 für lädt, 2 für status nicht verfügbar und 3 für wartet. Die Eigenschaft *responseText* enthält den Inhalt der Datei. Alles in einem sieht es dann in einer HTML-Datei so aus:

```
<html>  
<head>  
<title>AJAX</title>  
<script type="text/javascript"><!--  
var http = null;  
  
if (window.XMLHttpRequest) {  
    http = new XMLHttpRequest();  
} else if (window.ActiveXObject)  
{  
    http = new ActiveXObject("Microsoft.XMLHTTP");  
}
```

* siehe Abkürzungsverzeichnis

```

if (http != null) {
    http.open("GET", "datei.html", true);
    http.onreadystatechange = ausgeben;
    http.send(null);
}
function ausgeben() {
    if (http.readyState == 4) {
        document.getElementById("Ausgabe").innerHTML =
            http.responseText; }
}
//--></script>
</head>
<body>
HTML vom Server:
<div id="Ausgabe"></div>
</body>
</html>

```

Die Abbildung 1 [Abb.1] zeigt den Inhalt der Datei ‚datei.html‘ im Internet Explorer nach der Ausführung.



Abbildung 1: Inhalt der datei.html [7]

Die am häufigsten benutzten Methoden für das XMLHttpRequest-Object sind:

abort() laufende Anfrage wird angehalten
 send(..) Übergebene Daten werden gesendet
 getAllResponseHeaders() Satz Zeilensprung Bezeichnung wird zurückgeliefert
 getResponseHeader() einzelner Zeilensprung wird zurückgeliefert
 open(..) Zielangaben für die Übermittlung werden gesetzt
 setRequestHeader(..) Zeilensprung Bezeichnung wird zugewiesen

Die meist benutzten Eigenschaften für das XMLHttpRequest-Object sind:

status numerische Statusinformation
 statusText Statusinformation als Text
 readyState Objektstatus
 onreadystatechange Ereignishandler
 .responseText vom Server zurückgelieferter String (Inhalt)

2.1 Benutzte Sprachen und Methoden

Bei Ajax werden verschiedene bekannte Sprachen und Methoden eingesetzt, um arbeitsplatzoberflächeähnliche Webanwendungen zu realisieren. Einige von diesen sind:

- *Html*
- *DOM* zur Repräsentation der Daten oder Inhalte
- *Javascript* zur Manipulation des DOM und zur dynamischen Darstellung der Inhalte.
- *XMLHttpRequest* - Objekt, um Daten auf asynchroner Basis mit dem Webserver austauschen zu können.
- *REST* eine Technik für den Aufruf von Ressourcen
- *SOAP* - Übertragung von Methodennamen und Parametern als XML-Dokument
- *JSON* ein auf Javascript zugeschnittenes Format für Daten und Objekte

2.2 Vergleich mit herkömmlichen Webanwendungen

Wenn man bei einer Webanwendung ohne Ajax Technik ein Formular ausfüllt, werden die Daten an einen Server übermittelt und man bekommt eine neu generierte Webseite als Antwort zurück. Ajax Anwendungen schicken stattdessen Anfragen an den Server, bei denen nur Daten angefordert werden, die tatsächlich benötigt werden. Ein elementarer Bestandteil des Ajax-Konzeptes ist also die Veränderung des Applikationsmodells im Web bzw. der Art, in der Web-Anwendungen aufgebaut sind.

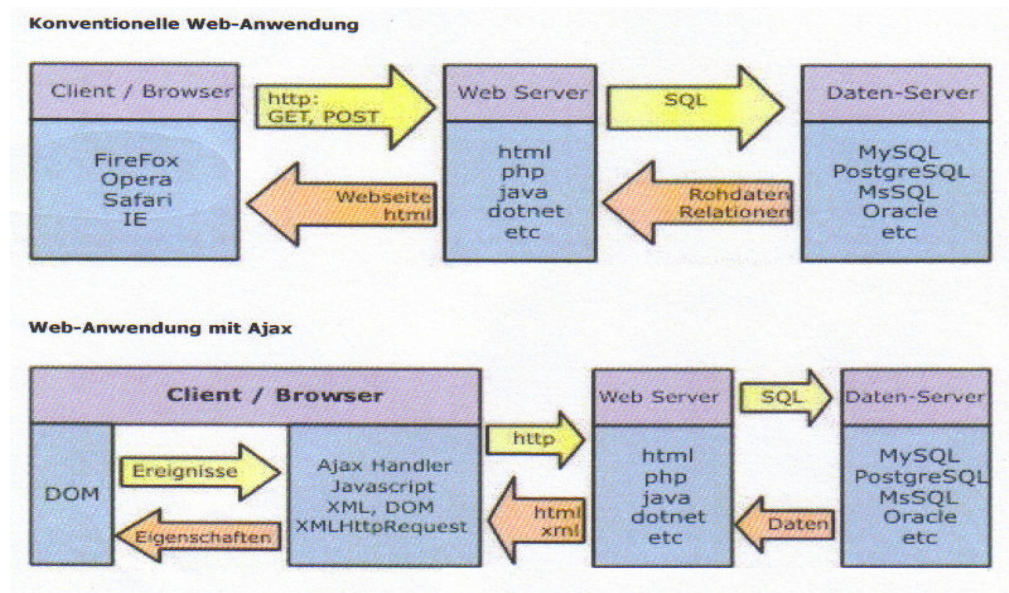
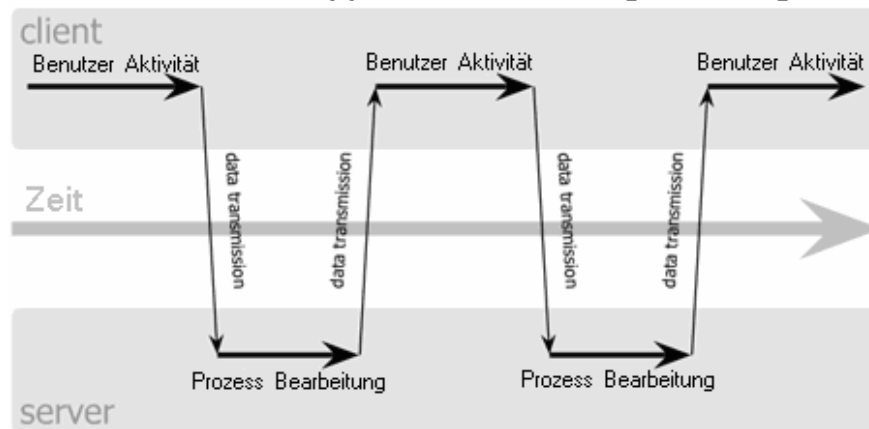


Abbildung 2: Konventionelle und Ajax basierte Webanwendung [9]

Während bei normalen Web-Applikationen nur zwei Instanzen miteinander agieren, der Benutzer (mit seinem Webbrowser) und der Server (mit der entsprechenden Software), kommt bei Ajax noch eine Ajax-Engine ins Spiel, die zwischen Benutzer und Server vermittelt. Dabei kann die Ajax-Software zum Beispiel schon die nächste Datei vom Server laden, während der Benutzer selbst noch gar nicht daran gedacht hat. Die üblichen Wartezeiten beim Aufruf einer neuen Webseite sollen so verringert oder sogar eliminiert werden. Außerdem ist es die Ajax-Engine, die die Präsentation (Rendering) der eigentlichen Daten übernimmt - es müssen also nur die wirklich relevanten Informationen vom Server übermittelt werden. Die Software kann dabei einfache Aufgaben ganz ohne Hilfe des Servers übernehmen. Weiterhin muß die Seite nicht für jede Änderung komplett aktualisiert werden. Im Endeffekt werden die Daten nicht direkt vom Browser verarbeitet, sondern mit Javascript-Prozessen zur Ajax-Engine weitergegeben. Als Ergebnis erhält man dann von der Ajax-Engine eine Benutzeroberfläche zurück, die viel schneller auf Eingaben reagiert.

klassisches Web Applikation Modell (gleichzeitig)



Ajax Web Applikation Modell (nicht gleichzeitig)

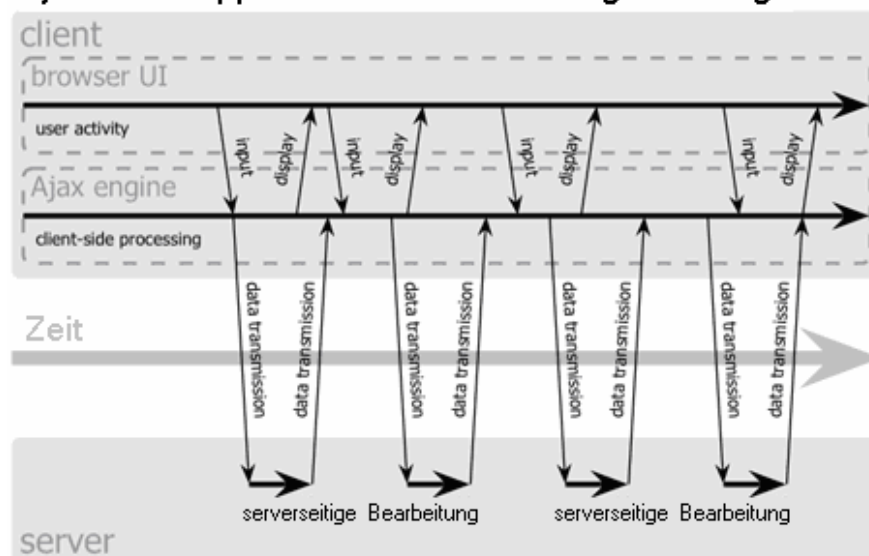


Abbildung 3: Klassisches und Ajax Web Applikationsmodell [8]

2.3 Client – Server Plattform

Ajax benutzt das Client - Server Architektur-Prinzip. Deswegen sind sowohl auf der Client-Seite bzw. im Browser als auch auf der Server-Seite eine Komponente für die Ajax-basierte Kommunikation notwendig. Auf der Client-Seite sind diese Javascript und XMLHttpRequest. Auf der Server-Seite sind es EJBs*, .NET* oder einfache Skript Komponenten. Mit deren Hilfe wird die Server-seitige Programmlogik umgesetzt. Sowohl die Programmlogik als auch die Anwendungslogik werden im Ajax-Kontext als Server-Plattform bezeichnet.

3 Beispiele für den Einsatz

Es gibt im Internet hunderte von Beispielen, in denen Ajax zum Einsatz kommt. Aber wenn man es auf die wichtigsten von diesen beschränkt, sind es Internetseiten, in denen Verwaltung- oder Formularaufgaben abgehandelt werden. Andere Einsatzgebiete sind zB. tief gestaffelte Baum-Navigtionen, Reservierungssysteme, schnelle Benutzer zu Benutzer Kommunikationen, Filterung von Daten sowie Online-Daten-Manipulationen, Autovervollständigung und Eingabe häufig gleicher Texte, Voting (Ja/Nein - Entscheidungen) bzw. Bewertungssysteme usw.

3.1 Anwendungen

- *Google Maps* ist mit Abstand eine der wohl berühmtesten Ajax Anwendungen, die mit Ajax Technologie entwickelt wurde. Es verknüpft eine interaktive Karte, Satellitenfotos, eine Navigationssoftware und die Google-Suche in einer einzigen Applikation. Wenn man in Google Maps zum Beispiel nach Stuttgart sucht, zeigt es auch alle Restaurants, die sich in Stuttgart befinden. Die Position der Markierungen wie zB. die Restaurants werden dem Browser vom Server mithilfe von Ajax Techniken übergeben und der Ajax Code im Browser setzt die Markierung entsprechend. Dabei können sogar Wegbeschreibungen von einem Punkt zum nächsten durch Ajax-Technik benutzt werden.
<http://maps.google.com/>
- *Gmail (Google Email)* ist ein gegenüber anderen Freemail-Diensten durch die Benutzung von Ajax-Technik viel schneller bekannt gewordener Email-Dienstleister. Der wesentliche Unterschied bei Gmail besteht im Funktionsangebot der browserbasierten Oberfläche, das sich an eigenständigen Mailprogrammen wie Outlook oder Thunderbird orientiert. Es umfasst ein Adressbuch, eine Rechtschreibprüfung sowie weitere Funktionen. Sie sind auch vergleichbar viel schneller und komfortabler ein lokal installiertes Mailprogramm, da durch die Ajax Technik die Funktionen größtenteils clientseitig umgesetzt werden.
<http://gmail.com/>
- *Ajax Trans* ist ein Übersetzungstool, das in erstaunlich hoher Qualität und in Echtzeit eingetippte Texte in fünf verschiedene Sprachen und wieder zurück übersetzen kann. Die Technik basiert wie bei anderen Beispielen auf Ajax und setzt entsprechend auch Javascript voraus. <http://ajax.parish.ath.cx/translator/>

* siehe Abkürzungsverzeichnis

- *SQL Designer* ist eine ebenso auf Ajax basierte Anwendung, die die Möglichkeit bietet, die SQL Anweisungen zur Erzeugung einer MySQL Datenbank mittels grafischer Oberfläche im Web zu erzeugen. Tabellen und Felder lassen sich einfach mit ein paar Mausklicks am Bildschirm erstellen. Ebenso einfach gestaltet sich die Definition von Relationen über Drag und Drop.
<http://ondras.praha12.net/sql/demo/>
- *Google Docs & Spreadsheets*, bezeichnet als „The Web Word Processor“, ist ein web-basierender Textverarbeitungsdienst, der den Anwendern ermöglicht, Dokumente in einem Html-Schnittstelle zu erstellen und zu editieren. Unter anderem werden Word-, RTF- und PDF Dokumente sowie Open-Document-Format unterstützt. Einige Funktionen wie zum Beispiel, dass mehrere Personen gleichzeitig an Dokumenten arbeiten können, ohne dass es zu Kollisionen oder Datenverlusten kommt oder dass Texte bequem verteilt oder im Blog veröffentlicht werden können, machen die Anwendung besonders interessant. Sowie bei anderen Anwendungen nutzt es auch das Ajax-Programmiermodell.
<http://docs.google.com/>
- *nexImage* ist ein Bildverarbeitungsprogramm, das die ortsunabhängige, schnelle, benutzerfreundliche und automatisierbare Bearbeitung von Bildmaterial auf einem Browser zur Verfügung stellt. Es ermöglicht eine Bildmanipulation ohne Installation auf dem Client und ohne komplizierte Lizenzierungsmodelle. Es eignet sich hervorragend zur Integration in bestehende CMS*-Lösungen und ergänzt deren Funktionsumfang mit einer bisher einzigartigen Technologie bzw. dem Ajax.
<http://neximage.com/>

3.2 Suchmaschinen

Die meisten Suchmaschinen ähneln sich in puncto Benutzeroberfläche sehr. Aber diejenigen unter ihnen, die die Ajax Technik benutzen, weisen zusätzliche Funktionen auf. Eine von diesen Funktionen ist die Echtzeiterkennung vom Text, den der Benutzer eintippt. Sobald man eine Buchstabe oder ein Wort eintippt, zeigt die Suchmaschine mögliche Suchvorschläge, die zu dem eingegebenen Text passen. Falls man zB. nach dem Wort ‚eine Fluggesellschaft‘ sucht und in der Suche noch delay (=Verzögerung) neben dem Suchbegriff schreibt, kommen automatisch links in der Ergebnisliste zu den Seiten mit Texten, Informationen über die verzögerte Auslieferung des Flugzeuges dieser Fluggesellschaft. Diese Suchmaschinen bauen darauf, auf dass die Anwender gerne gleich einen visuellen Eindruck vom Suchergebnis erhalten, um weiter ausfiltern zu können. Einige von diesen Suchmaschinen sind:

- snap.com <http://snap.com>
- Alltheweb.com <http://alltheweb.com>
- live.com <http://live.com>
- wikiwax.com <http://wikiwax.com>
- pixsy.com <http://pixsy.com>
- google suggest <http://www.google.com/webhp?complete=1&hl=en>

* siehe Abkürzungsverzeichnis

3.3 Webbrowser

Für Ajax ist Javascript genau das gleiche wie für uns Menschen die Luft. Das bedeutet ohne Javascript kann Ajax nicht leben. Da Javascript eine clientseitige Programmiersprache ist, müssen die Webbrowser es auch unterstützen. Die Javascript-Programme werden im Webbrowser ausgeführt. Dies wird dadurch möglich, dass Javascript in Html eingebettet wird. Html-Dateien können also Javascript-code erhalten, wenn sie im World Wide Web unterwegs sind. Inzwischen unterstützen aber die meisten gebräuchlichen Webbrowser Javascript. Diese sind:

- Internet Explorer
- Mozilla Firefox
- Opera
- Apple Safari
- Netscape
- Konqueror

3.4 Entwicklungsumgebungen

Ajax Anwendungen von Hand zu entwickeln ist mit viel Arbeit verbunden. Hier helfen die Frameworks bzw. Entwicklungsumgebungen. Ein Framework ist ein Programmgerüst oder ein Skelett, der von einem Programmierer eingebauten Implementierungen bzw. den Ablauf einer Anwendung beinhaltet. Sie werden mit einem allgemeinen Ziel, einer Wiederverwendung bekannter Muster entwickelt und genutzt.

- Das bekannteste Ajax-Framework, auch genannt "the Javascript Toolkit" ist *Dojo*. Sie ist ein Vertreter der rein clientseitigen Frameworks und hat den Vorteil, mit jeder Servertechnologie zu funktionieren. Gesponsert wird Dojo von IBM und AOL, womit die Zukunft von Dojo sehr gut aussieht.
<http://dojotoolkit.org/>
- Ein anderes Framework für Ajax ist *Atlas* von der Firma Microsoft. Im Mittelpunkt von Atlas stehen die vorgefertigten Komponenten, die dem Entwickler viel Handarbeit ersparen sollen. Daneben gibt es noch eine eigene Javascript-Bibliothek und die Integration von Webservices.
<http://atlas.asp.net>
- *Spry*, ein Ajax Framework von der Firma Adobe, besitzt die Besonderheit, dass Html weiterhin die Kernkomponente bleibt und geringe Javascript Kenntnisse ausreichen, um mit diesem Framework umzugehen. Laut Adobe soll sich Spry in Kombination mit jeder beliebigen Webdesign-Anwendung einsetzen lassen und richtet sich an Nutzer, die von Ajax profitieren können. Ein schönes Beispiel von diesem Framework kann man auf der folgenden Internetseite betrachten:
<http://labs.adobe.com/technologies/spry/demos/gallery/index.html>

4 Probleme

Der Einsatz der AJAX-Programmiertechnik bringt auch Probleme mit sich. Bereits ohne AJAX ist die Komplexität in einer Client-Server-Umgebung erheblich. Es muss in der Regel gleichzeitig auf Client und auf Server programmiert werden. Meist sind dabei viele Programmiersprachen im Einsatz. Der Datenbankserver kommuniziert in SQL*, auf dem Webserver kommen Sprachen wie PHP*, Perl* und Java auf unterschiedlichen Plattformen wie J2EE* oder .Net zum Einsatz, die Darstellung auf dem Client wird per Html und CSS* definiert. Beim Einsatz von AJAX wird auch noch auf dem Client programmiert. Das heißt der Entwickler muss sich zusätzlich mit Konzepten wie DOM*, XML und JavaScript befassen.

4.1 Vorteile und Nachteile

Ein Vorteil der Ajax Technologie ist, dass Daten verändert werden können, ohne dass die komplette Seite vom Browser neu geladen werden muss. Dadurch wird der Verkehr im Internet auch reduziert, da es vermieden wird, dass statische Daten, die sich unter Umständen nicht ändern, fortwährend über das Internet übertragen werden. Ein anderer Vorteil ist die Zeitersparnis. Ein Beispiel dazu ist die fortgeschrittene Benutzungsoberfläche. Während der Benutzer ein Formular ausfüllt, können gleichzeitig Aktualisierungen vorgenommen werden, komplexe Daten können direkt bestätigt werden oder es können im Hintergrund auf dem Server Berechnungen durchgeführt werden. Dadurch verkürzen sich auch die Ladezeiten. Da Ajax freie Webtechnologien verwendet ist es auch kostenlos gegenüber anderen Techniken. Ajax hat eine große Nutzergruppe. Sie wird von vielen Menschen weiterentwickelt, dadurch wird der Einsatz komfortabler und umfangreicher, wie man es von den Frameworks sehen kann. Durch die clientseitige Bearbeitung der meisten Aufgaben und da seltener eine Verbindung zum Server hergestellt muss, wird auch die Serverlast gemindert. Und der wichtigste Vorteil von allen ist, dass Ajax gegenüber anderen Technologien wie Shockwave oder Flash keine zusätzlichen Browser -Plugins benötigt.

Wie jede andere Technologie hat auch Ajax einige Nachteile, die aber mit der Zeit sicher behoben werden können. Einer von diesen ist die Verwendung des Zurück-Knopfs. Er funktioniert nicht mehr effektiv, da die dynamischen Ajax Schritte nicht in der Geschichte des Browsers gespeichert werden. Andere Nachteile wären zB., dass Ajax auf Javascript basiert, was in manchen Browsern ausgeschaltet sein kann, womit dann die Funktionen hinfällig sind oder dass bestimmte Zustände nicht als Lesezeichen abgespeichert werden können oder dass man auch selbst in seine Anwendungen visuelle Rückmeldung wie die Sanduhr integrieren muss, um die Benutzerfreundlichkeit zu erhöhen.

4.2 Probleme und Lösungen

Ein großes Problem bei der Ajax Technik, genannt *Polling Problem*, ist, dass der Ajax Client permanent beim Webserver nachfragen muss, ob ein neues Ereignis erzeugt wurde. Das erzeugt eine andere Auslastung auf dem Server, verglichen mit der Last, die von herkömmlichen Anwendungen erzeugt wird. Die Lösung für dieses Problem ist, die Poll-Anfragen solange zurückzuhalten, bis ein tatsächliches Ereignis eintritt. Deswegen ist bei Ajax Anwendungen im Idealfall eine serverseitige Anfrage verknüpft, die dem Client eine Antwort schickt, falls ein Ereignis stattfindet.

* siehe Abkürzungsverzeichnis

Ein anderes Problem entsteht durch die Lösung dieses Polling Problems. Bisher war es üblich, für jede Anfrage an den Server einen Thread zu erzeugen, dessen Ressource sofort nach dem Abarbeiten der Anfrage wieder freigegeben werden konnten. Bei der Polling Technik ist diese Freigabe des Threads jedoch nicht möglich. Man kann dieses Problem aber durch Verwendung eines Anwendung Servers, der das Prinzip der Fortsetzung unterstützt, aus der Welt schaffen.

5 Zusammenfassung

Trotz einiger Probleme wird Ajax ein immer beliebteres Mittel um Webseiten interaktiver zu gestalten. Mit Ajax lassen sich Webseiten leichter und intuitiver nutzen, teilweise kann man kaum einen Unterschied zu installierten Arbeitsplatzanwendungen ausmachen. Es wird vielleicht Flash nie ersetzen können, als Erweiterung und Verbesserung kann es aber von großem Nutzen sein. Es wird sich weiterentwickeln, so wie die Browser auch. Die größte Herausforderung bei der Entwicklung von Ajax-Anwendungen für die Designer wird sein, zu vergessen, was sie über die Beschränkungen des Web zu wissen glauben und sich einen größeren, reichhaltigeren Bereich der Möglichkeiten vorzustellen. Ajax und all die verwandten Möglichkeiten werden uns in Zukunft noch sehr oft begegnen, wir sehen heute erst die Spitze des Eisberges.

Abkürzungsverzeichnis

AJAX	Asynchronous JavaScript and XML
XML	Extensible Markup Language
HTML	Hypertext Markup Language
DHTML	Dynamic HTML
CSS	Cascading Style Sheets
DOM	Document Object Model
REST	ReStructuredText
SOAP	Simple Object Access Protocol
JSON	JavaScript Object Notation
EJB	Enterprise JavaBeans
.NET	Network Entity Title
SQL	Structured Query Language
RTF	Rich Text Format
PDF	Portable Document Format
CMS	Content-Management-System
PHP	Personal Home Page Tools
J2EE	Java Platform, Enterprise Edition

Literaturverzeichnis

- [1] Buch: Javascript und Ajax
Kapitel 18 Ajax
Galileo Computing

- [2] Bericht: Morgenluft für Web-Anwendungen
InfoWeek Online Archiv
http://www.infoweb.ch/archive/ar_single.cfm?ar_id=16719&ar_subid=3&sid=0
Abgerufen am 04.11.06

- [3] Bericht: Ajax (Programmierung)
http://de.wikipedia.org/wiki/Ajax_%28Programmierung%29
Abgerufen am 04.11.06

- [4] Webseite: Prologon AG
<http://w3.prologon.ch/?go=901>
Abgerufen am 04.11.06

- [5] Webseite: NetWiki – Ajax
<http://www.net-wiki.de/index.php?title=AJAX>
Abgerufen am 04.11.06

- [6] Webseite: Ajax Quellensammlung
<http://www.drweb.de/weblog/weblog/?p=454>
Abgerufen am 04.11.06

- [7] Bild aus dem Buch Javascript und Ajax
Kapitel 18 Ajax
Galileo Computing

- [8] Bild aus der Webseite TwoZero
Uni Köln
http://twozero.uni-koeln.de/content/e174/index_ger.html
Abgerufen am 04.11.06

- [9] Bild aus der Seite Prologon AG
<http://w3.prologon.ch/?go=901>
Abgerufen am 04.11.06