



7363 - Web-basierte Anwendungen

Eine Vertiefungsveranstaltung
mit Schwerpunkt auf XML-Technologien



AJAX

Asynchronous JavaScript and XML



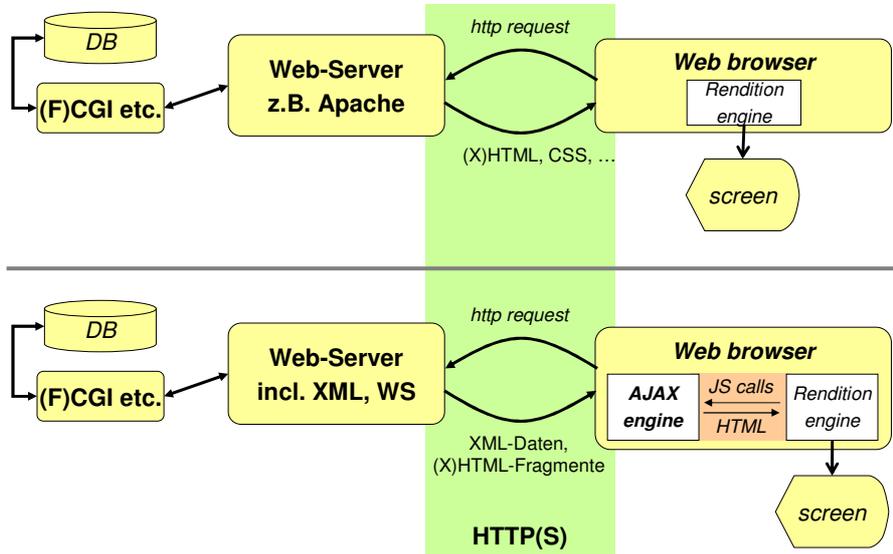
Einführung

Interaktivere Benutzerschnittstellen im Web
mit Ajax

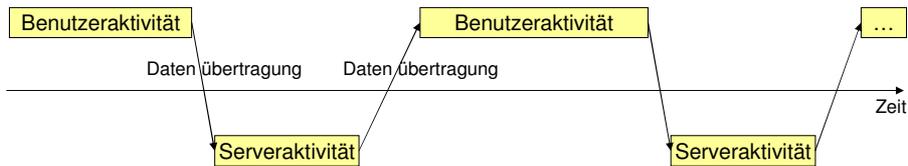


- Ajax – Eine Mischung bekannter Techniken:
 - Präsentation von Information auf der Basis von Standard, insbesondere von **XHTML** und **CSS**
 - Dynamische Anzeigen und Interaktion mit den Inhalten mittels **DOM** (*Document Object Model*)
 - Datenrepräsentation und –transformation mit **XML** and **XSLT**
 - Asynchroner Datenaustausch mit **XMLHttpRequest**
 - und **JavaScript**, um all dies zu verbinden

WBA: Ajax vs. traditionelle Interaktionen



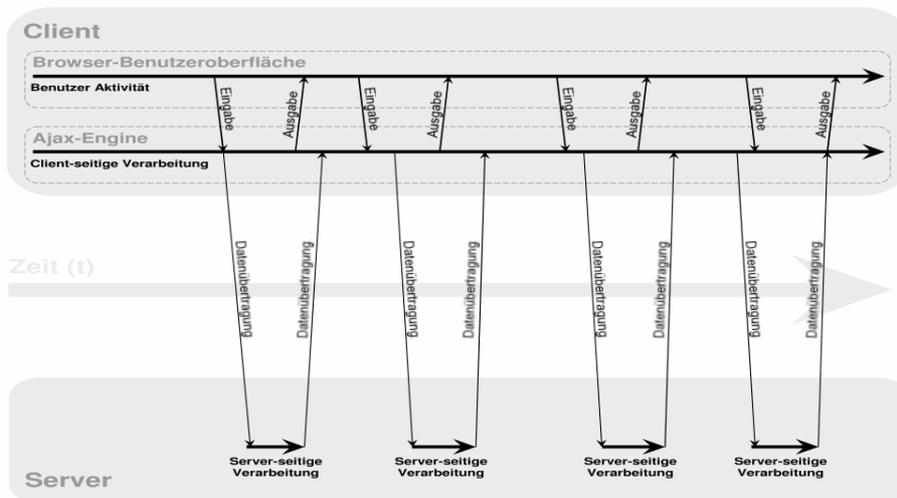
Ajax: Synchrone Datenübertragung



Traditionelle WBA wechseln zwischen Benutzer- und Server-Aktivitäten

Anwender empfinden die entstehenden Wartezeiten als störende Unterbrechungen ihres Arbeitsflusses.

Ajax-Modell einer Web-Anwendung (asynchrone Datenübertragung)



Quelle: de.wikipedia.org

Ajax: Alternative Implementierungen

- Direkte Ajax-Implementierung
 - Client besitzt API zur XML-basierten Kommunikation mit dem Server (XMLHttpRequest, evtl. SOAP)
 - Datenaustausch effizient und flexibel,
 - Transformation erforderlich, komplexer *client*
- Indirekte Ajax-Implementierung
 - Client tauscht HTML-Fragmente mit Server aus
 - Client aktualisiert Darstellung mittels DOM
 - Beispiel: Einfügen / Aktualisieren von Listeneinträgen, Positionen einer Bestellung etc.
 - Implementiert in Rails

- Beispielanwendungen

- *Google Maps*: <http://maps.google.de>

- „Gleitende“ Verschiebung des Sichtbarkeitsfensters auf die Karte,
- vorausschauendes Nachladen der nächsten Kacheln
- Ähnliche Wirkung mit Java Applets: Stadtplan Wiesbaden.
- Gegenbeispiel: <http://stadtplan.frankfurt.de>

- *Google Suggest*: <http://www.google.com/webhp?complete=1&hl=en>

- Sofort angebotene Auswahlliste von Suchbegriffen allein aufgrund der bisher eingetippten Zeichen.

- **XMLHttpRequest: Details**

- Aktuelle Quelle: W3C-Entwurf vom 5. April 2006,
<http://www.w3.org/TR/XMLHttpRequest/>

- Code-Beispiel:

```
var xmlhttp = new XMLHttpRequest();
if (xmlhttp) {
    xmlhttp.open('GET', 'beispiel.xml', true);
    xmlhttp.onreadystatechange = function () {
        if (xmlhttp.readyState == 4) {
            alert(xmlhttp.responseText);
        }
    };
    xmlhttp.send(null);
}
```

- Kleine JS/DOM-Demo

cd Ajax-XMLHttpRequest

```
«interface»
XMLHttpRequest

- onreadystatechange: Function
- readyState: short
- responseText: DOMString
- responseXML: Document
- status: short
- statusText: DOMString

+ abort() : void
+ getAllResponseHeaders() : DOMString
+ getResponseHeader(DOMString) : DOMString
+ open(DOMString, DOMString) : void
+ open(boolean, DOMString, DOMString) : void
+ open(DOMString, boolean, DOMString, DOMString) : void
+ open(DOMString, DOMString, boolean, DOMString, DOMString) : void
+ send(DOMString) : void
+ send(Document) : void
+ setRequestHeader(DOMString, DOMString) : void
```

Quelle: de.wikipedia.org

• Problemgebiete

– *Back button*, Lesezeichen

- Durch das dynamische Verhalten von AJAX-Anwendungen funktionieren diese Browser-Elemente nicht mehr.
- Auswege:
 - Beschränkung von AJAX-Funktionen auf (kleine) Funktionsgruppen, Verwendung von *back button* und Lesezeichen zwischen den Gruppen, Verzicht auf diese Elemente innerhalb einer Gruppe.
 - Verlagern von Ajax- Aktivitäten in unsichtbare „iframes“ in statischer HTML-Seite

– *Polling-Problem*

- Web Server unterliegen dem C/S-Modell – sie können den Client nicht „zurückrufen“!
- Asynchrones Verhalten des XMLHttpRequest-Objekts wird durch Nebenläufigkeit (*multi-threading*) erreicht. Diese zusätzlichen Threads existieren länger als bei normalen C/S-Anfragen und binden Ressourcen auf Client-Seite entsprechend länger.
- Clients können durch ungeschicktes Vorgehen (häufiges *polling*) neue, erhebliche Serverlasten verursachen. Auch kann sich die Anzahl gleichzeitig offener TCP-Verbindungen des Servers erhöhen.

Ajax und SOAP

10

- Ajax und SOAP?
 - ... sollten doch ideal zueinander passen ☺
- Lösungsansatz:
 - SOAP-Dokument („*envelope*“) wird *client*-seitig erzeugt.
 - Es wird mittels XMLHttpRequest an einen WS-Provider *per http binding* übertragen.
 - Aus Sicht des WS *providers* handelt es sich um eine normale WS *consumer*-Anfrage, die mit einem geeigneten *envelope* beantwortet wird.
 - Client-seitig ist dann zwischen SOAP und der JavaScript-Ebene zu vermitteln.
 - Diese Vermittlung JS \leftrightarrow SOAP überlässt man Bibliotheksfunktionen, z.B. denen von „ws.js“ von IBM:
 - <http://www-128.ibm.com/developerworks/webservices/library/ws-wsajax/>



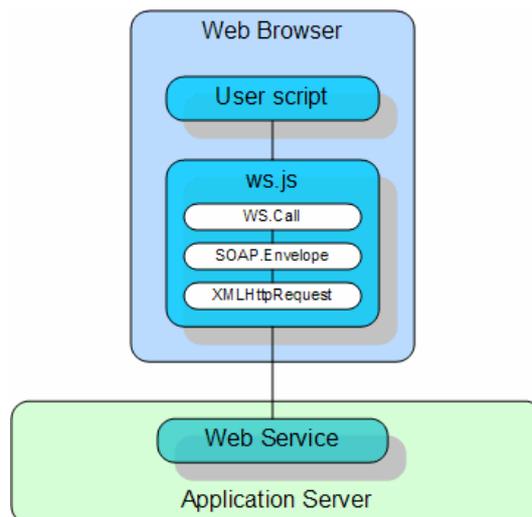
21.06.2006

H. Werntges, FB Design > Informatik > Medien, FH Wiesbaden

13

Ajax und SOAP

10



Quelle: <http://www-128.ibm.com/developerworks/webservices/library/ws-wsajax/>

21.06.2006

H. Werntges, FB Design > Informatik > Medien, FH Wiesbaden

14



Ajax: Quellen



- [1] [http://de.wikipedia.org/wiki/Ajax_\(Programmierung\)](http://de.wikipedia.org/wiki/Ajax_(Programmierung))
– Sehr kompetenter, gut verständlicher und weiterführender Übersichtsartikel!
- [2] Jesse James Garrett: *Ajax: A New Approach to Web Applications*. Adaptive Path LLC, 18. Februar 2005, <http://www.adaptivepath.com/publications/essays/archives/000385.php>
– Der Artikel, der die Bezeichnung AJAX nachhaltig prägte.
- [3] <http://www.openajax.net/>
– Beispiel- und Linksammlung
- [3] Drew McLellan: *Very Dynamic Web Interfaces*. 9. Februar 2005, <http://www.xml.com/pub/a/2005/02/09/xml-http-request.html>
– Ein Artikel u.a. mit Details zum Umgang mit XMLHttpRequest



Ajax im Projekt?



- Mögliches Einsatzgebiet
 - *Ergebnislisten-Interface, Suche nach Name*
 - Eingabefeld für Name, „Suchen“-Button
 - Tabelle oder Textfeld mit den ersten ca. 10 Treffern
 - Mit jedem eingetippten Zeichen wird die Trefferliste aktualisiert:
 - Anzeige = Die ersten 10 Namen, die mit den eingetippten Zeichen beginnen!
 - *Ergebnislisten-Interface, Suche nach Verein/Ort*
 - Analog zur Namenssuche
- Hinweise
 - Rein als Anregung zu verstehen, keine Sonderwertung, kein Pflichtpunkt!
 - Realisierung sollte mit den Ajax-Möglichkeiten von **Rails** möglich sein.
 - Indirekte Ajax-Implementierung genügt hier (Einfügen auf HTML-Ebene).
 - Selbst die großen Stadtmarathon-Seiten bieten diesen Komfort bisher nicht!