

Web 2.0

Stefan Schöttelndreyer

Matr. Nr. 652351

Zusammenfassung: *Web 2.0 beschreibt den aktuellen Wandel des World Wide Web. Der Begriff impliziert eine Sammlung von Techniken, mit deren Hilfe Web-Anwendungen noch intuitiver und interaktiver gestaltet werden können.*

1 Einführung

Das Internet und die durch es zur Verfügung gestellten Dienste sind aus unserem Leben kaum noch wegzudenken. Seit einigen Jahren ersetzen E-Mails mehr und mehr Faxe und Briefe, Telefonate werden über das Internet geroutet und Nachrichten werden online gelesen. Es gibt kaum etwas, das nicht auch im Web erledigt werden kann. Vielleicht genau deswegen ist das Netz der Netze im stetigen Wandel. Ein Teil des E-Mail Verkehrs wird durch Instant Messaging ersetzt, Mailinglisten verlagern sich in Foren und sogar das Verhalten von Webseiten ändert sich durch immer neue Technologien. Große Softwarehersteller haben das Potenzial des WWW erkannt und arbeiten aktiv an der Verbesserung dessen Benutzerfreundlichkeit. In den folgenden Kapiteln werden einige der Techniken näher erläutert, die das Web zu dem machen was es heute ist. Einige dieser Techniken haben ihren Ursprung in den Grundzügen des Internets, andere sind Neuentwicklungen um den veränderten Bedürfnissen gerecht zu werden. Visionäre sehen das Internet und seine Dienste dank steigender Bandbreite der Zugänge und neuer Technologien immer mehr mit dem klassischen Desktop verschmelzen. Sogar vom "Netz als Betriebssystem" ist die Rede. [30]

2 Die Geschichte des World Wide Web

Zu Beginn des World Wide Web standen die Inhalte und nicht etwa die Darstellung der Informationen im Vordergrund. Um nun die übermittelten Daten logisch beschreiben zu können, entwickelte der Brite Tim Berners-Lee 1989 am CERN in Genf ein Format zur Repräsentation dieser Daten – eine auf SGML basierende Auszeichnungssprache, die später den Namen "HyperText Markup Language" erhielt. Auch der erste Webbrowser wurde im Jahr 1990 von Berners-Lee, damals noch unter dem Namen "WorldWideWeb", vorgestellt. Ende des Jahres 1993 erschien mit dem Namen "Mosaic" der erste Browser, der außer Text auch Grafiken auf einer Seite anzeigte. [28]

Die Fähigkeit das Layout von Webseiten mittels HTML zu definieren wurde erst später durch Softwarehersteller geprägt, die ihre Vorherrschaft im Bereich Web-Technologien ausbauen wollten. Um bestimmte Layouteffekte zu erzielen, vermischten Webentwickler häufig Tabellen, HTML-Elemente und eingebettete Grafiken. Es entstand die sogenannte "Tag Soup", eine wirre Aneinanderreihung von Tags, die von den Fehlertoleranzmechanismen der Browser meist unterschiedlich interpretiert wurden. Die Nachteile dieser Vorgehensweise liegen klar auf der Hand; die Darstellung der Webseiten ist abhängig von Browser oder Betriebssystem, benachteiligte Personengruppen, die beispielsweise auf Screen Reader angewiesen sind werden ausgegrenzt, et cetera. [11]



Abbildung 1: Suchaufkommen des Begriffs "Web 2.0"

Um die wachsende Menge von Seiten effizient bearbeiten und verwalten zu können wurden Content-Management-Systeme (CMS) entwickelt, die datenbankbasiert Seiten dynamisch zur Laufzeit generieren. Die Arbeit an großen Seiten von mehreren Autoren wurde durch eine komplexe Benutzerverwaltung erleichtert und der automatisch generierte Code entspricht immer öfter gültigen Webstandards. Wollte man tatsächlich eine Versionierung im Web vornehmen, so müsste diese Technik mit der Versionsnummer 1.5 bedacht werden.

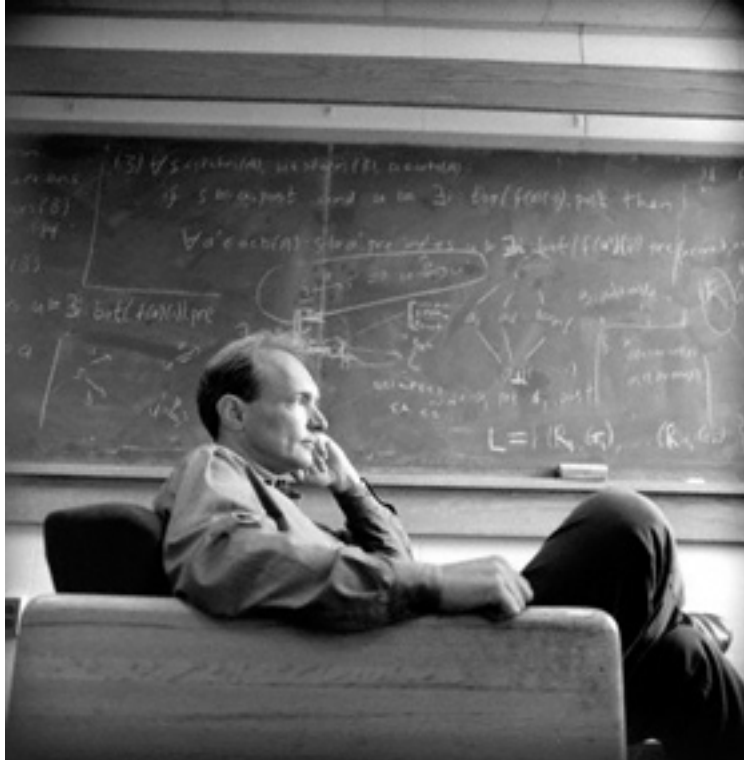


Abbildung 2: Tim Berners-Lee am MIT, 2000

Seit dem Jahr 2005 kündigt sich eine Renaissance im World Wide Web an. Die Gemeinschaft der Webanwender empfindet ein Gefühl der Veränderung von Regeln und Geschäftsprozessen. Der Internet-Browser wird zunehmend als universelle Benutzerschnittstelle für jegliche Art von Anwendungen genutzt, traditionelle Desktop-Anwendungen aktualisieren sich selbstständig über das Netz, die Datenhaltung verlagert sich zunehmend vom PC ins Internet und selbst die im Web 1.0 so passiven und konsumierenden Benutzer werden aktiv. Mittels Wikis, Foren und Blogs verschmelzen die klassischen Grenzen zwischen Editoren und Nutzern. Diese, aber auch einige andere Umstände werden als Grundstein für das Verständnis des Web 2.0 angesehen, wobei darauf hingewiesen werden muss, dass keine anerkannte Definition existiert, welche Technologie Web 2.0 zugehörig ist. [26]

3 Die Technik des Web 2.0

In den folgenden Abschnitten werden einige der Techniken näher betrachtet, die gemeinhin unter den Sammelbegriff Web 2.0 fallen. Viele dieser Techniken sind keinesfalls Neuentwicklungen. Die erste Version von XML wurde schon im Februar 1998 vom W3C vorgestellt. JavaScript existiert schon seit 1995. Die ständige Erweiterung und intelligente Verknüpfung der zur Verfügung stehenden Techniken bilden den Grundstein für das neue Web.

3.1 (X)HTML und CSS

Der Aufbau von HTML-Dokumenten ist denkbar einfach: Jedes Element setzt sich aus einem Start- und einem Endtag zusammen. Das Starttag kann zusätzlich durch Attribute ergänzt werden, die das Element näher beschreiben. Der Inhalt eines Elements steht zwischen Start- und Endtag, es kann reiner Text oder wiederum ein Element sein. Diese hierarchische Anordnung von Elementen kann auch als Baumstruktur visualisiert werden, was weithin der DOM-Repräsentation (s. Kapitel 3.3) entspricht. XHTML (Extensible

Hypertext Markup Language) ist eine Neuformulierung von HTML 4 in XML 1.0, d.h. als Sprachgrundlage dient nicht mehr SGML (Standard Generalized Markup Language) sondern die SGML-Teilmenge XML (Extensible Markup Language), auf die in Kapitel 3.4 näher eingegangen wird. HTML 4.01 ist vollständig aufwärtskompatibel zu XHTML 1.0, d.h. Dokumente im HTML 4.01-Standard können verlustfrei in XHTML 1.0 umgeformt werden, da XHTML 1.0 alle Elemente von HTML 4.01 enthält. Die sich momentan noch in Entwicklung befindliche Version 2.0 des XHTML-Standards wird nicht mehr kompatibel zum HTML-Standard sein. [12, 13, 15, 19]

Um der strikten Trennung von Strukturinformationen und Layoutangaben gerecht zu werden, benötigt man eine weitere Technik, die in der Lage ist, Dokumentstrukturen durch Stilinformatoren anzureichern. Genannt wird diese Technik Cascading Style Sheets oder CSS. Mit CSS ist es möglich, Eigenschaften von bestimmten Objekten anzugeben ohne in die Strukturinformation des Dokuments einzugreifen. CSS kann innerhalb von (X)HTML-Tags, als internes Stylesheet in einem (X)HTML-Dokument oder als externes Stylesheet definiert sein, wobei die externe Integration die am häufigsten Verwendete darstellt. Sie hat den Vorteil, dass man bei mehreren Seiten mit gleichem Erscheinungsbild nur eine zentrale Stylesheet-Definition benötigt und sich z.B. das Layout ohne Änderung am (X)HTML-Dokument modifizieren lässt. Eine CSS-Regel besteht aus einem Selector zur Auswahl der anzuwendenden Objekte und einem Rumpf, in dem die neuen Eigenschaften der Objekte aufgeführt werden. Es ist zudem möglich, mehrere Regelsätze auf ein Dokument anzuwenden. Auf diese Weise können Eigenschaften von Objekten überschrieben werden. [12]

3.2 JavaScript

Die erste Version der Scriptsprache wurde von Brendan Eich entwickelt und Ende 1995 mit Netscapes Browser Version 2.0B3 unter dem damals ins Produktkonzept von Netscape passenden Namen LiveScript eingeführt. Um den Java-Boom der damaligen Zeit auch für die Scriptsprache LiveScript nutzen zu können, gaben Netscape und Sun noch vor 1996 die Umbenennung der Sprache in JavaScript bekannt. Bei ihrer Einführung konnte die Sprache nicht viel mehr als z.B. das Überprüfen von Formulareingaben oder die dynamische Generation von Inhalt während des Ladens einer Webseite. Dies änderte sich 1997, als Microsoft HTML, Scripting und Style Sheets zu einer Technik mit Namen DHTML zusammenfasste. Nun war es möglich, Seiteninhalte während der Darstellung im Browser zu modifizieren. Diese Eigenschaft macht JavaScript zu einem wichtigen Bestandteil von Ajax. [2, 11]

Wie anfangs schon erwähnt ist JavaScript eine Scriptsprache. Der Hauptunterschied zu Programmiersprachen wie Java oder C++ besteht darin, dass sie immer in einer bestimmten Umgebung ablaufen. Bei JavaScript ist dies der Webbrowser. Wollte man in Java eine Dialogbox erstellen, so müsste man sie selbst entwerfen und die Funktionalität dieser Dialogbox programmieren. JavaScript hingegen benutzt die vom Browser vorgegebenen Dialogboxen.

A scripting language is a programming language that is used to manipulate, customise, and automate the facilities of an existing system. In such systems, useful functionality is already available through a user interface, and the scripting language is a mechanism for exposing that functionality to program control. In this way, the existing system is said to provide a host environment of objects and facilities, which completes the capabilities of the scripting language. A scripting language is intended for use by both professional and non-professional programmers. To accommodate non-professional programmers, some aspects of the language may be somewhat less strict.¹

Um den Streit zwischen Microsoft und Netscape um die Weiterentwicklung von Scriptsprachen wie JavaScript bzw. JScript zu schlichten, standardisierte die private Normungsorganisation Ecma International erstmals im Juni 1997 die Sprache ECMAScript, die im April 1998 zusätzlich als ISO-Standard erschien (ISO/IEC 16262). Um die Kompatibilität zu ECMAScript zu gewährleisten, muss eine Scriptsprache die Grundfunktionen von ECMAScript unterstützen. Die konkrete Syntax der Sprache ist hierbei unerheblich. Sowohl JScript als auch JavaScript ab der Version 1.5 erfüllen die Kompatibilitätsbedingungen. [11]

1 ECMAScript Language Specification, Standard ECMA-262, 3rd edition, Dezember 1999, S. 1

JavaScript kann auf verschiedene Arten in ein (X)HTML-Dokument eingebunden werden:

- unmittelbar in dem Dokument zwischen den Tags `<script>` und `</script>`
- in einer externen Datei, deren Pfad im Header des (X)HTML-Dokuments eingebunden wird
- innerhalb von (X)HTML Tags

Bei der Deklaration einer Variablen in JavaScript steht der Typ der Variablen noch nicht fest. Es ist sogar möglich, dass der Typ einer Variablen bei der Programmausführung wechselt. Sprachen mit diesen Eigenschaften nennt man auch untypisiert.

JavaScript eignet sich unter anderem auch deshalb so gut zur dynamischen Webseitengestaltung, weil es auf das Document Object Model (s. Kapitel 3.3) zugreifen kann und so die Möglichkeit besteht, Elemente eines Dokuments zur Laufzeit zu ermitteln und zu modifizieren. Einige Beispiele gibt die folgende Tabelle:

<code>getElementById("Identifizier")</code>	<code><p id="Identifizier">Absatz...</p></code>
<code>getElementsByName("Name")</code>	<code><input name="Name" type="text"></code>
<code>getElementsByTagName("p")[0]</code>	<code><p>Absatz...</p></code>

Tabelle 1: Zugriff auf Elemente im DOM

Die CSS Eigenschaften der über die oben gezeigten Methoden angesprochenen Elemente im Elementknotenbaum des DOM können mittels des `style`-Objekts beliebig modifiziert werden.

Ein weiterer wichtiger Bestandteil von JavaScript sind Event-Handler. Die Notation von Event-Handleern erfolgt meist als Attribut im (X)HTML-Tag. Die Bezeichnungen der Event-Handler beginnen mit "on" und beschreiben ein bestimmtes Ereignis. So steht zum Beispiel "onmouseover" für das Überfahren des Elements mit der Maus. Der Wirkungsbereich eines Event-Handlers ist immer genau das Element, dessen Attribut den Event-Handler enthält. Ist beispielsweise ein Bild folgendermaßen in ein XHTML-Dokument eingebunden: ``, so wird die vorher an anderer Stelle definierte Funktion `doSomething()` nur ausgeführt, wenn der Benutzer mit dem Mauszeiger über den Bereich des Bildes fährt. [12]

JavaScript verfügt über eine Menge an Kontrollstrukturen wie Schleifen und bedingte Anweisungen, auf die hier nicht weiter eingegangen wird. Der Ablauf eines JavaScript-Programms ist, wird er nicht durch Funktionsaufrufe oder Kontrollstrukturen beeinflusst, sequenziell, d.h. nacheinander Anweisung für Anweisung von oben nach unten.

3.3 Das Document Object Model

Das Document Object Model wurde 1998 erstmals vom W3C unter dem Namen "DOM Level 1" standardisiert. Es handelt sich hierbei um eine plattformunabhängige Programmierschnittstelle, die den Zugriff auf XML- und HTML-Dokumente realisiert. Vorläufer des DOM war das "Dynamic HTML Model", ein nicht standardisiertes und proprietäres Modell, mit dem Browserhersteller eine Schnittstelle zwischen Scriptsprachen und HTML-Dokumenten herstellten, um diese dynamisch zu manipulieren. In der Literatur wird in diesem Zusammenhang häufig vom "DOM Level 0" gesprochen. Die momentan aktuellste Spezifikation des W3C ist DOM Level 3, jedoch wird dieser Standard noch von keinem Browser vollständig unterstützt. In Folgenden werden die wichtigsten Webbrowser mit ihren jeweiligen DOM-Level Unterstützungen aufgeführt:

Konqueror 3.4	DOM Level 1, 2, 3 teilweise
Microsoft Internet Explorer 6	DOM Level 1
Mozilla Firefox 1.5	DOM Level 1, 2, 3 teilweise
Opera 8	DOM Level 1 teilweise, 2 teilweise

Tabelle 2: Umsetzung von DOM-Standards aktueller Browser

Ausgezeichnet werden die Daten in XML wie auch in anderen Auszeichnungssprachen mit Tags. Tags beinhalten Informationen über die von ihnen eingeschlossenen Daten, sind also sogenannte Metadaten. Im Gegensatz zu HTML gibt es in XML keinen endlichen, bestimmten Satz von Tags für die Strukturierung eines Dokuments. Es können also Tags selbst definiert werden, um die Struktur oder Semantik von Elementen eines Dokuments zu bestimmen. XML kann also eher als Satz von Regeln verstanden werden, mit denen es möglich ist, eigene Auszeichnungssprachen zu kreieren, die in selbst definierten Tags angeordnet sind und aus selbst definierten Strukturen bestehen. [1, 6, 14]

XML-Dokumente sind ähnlich wie HTML- oder XHTML-Dokumente (s. Kapitel 3.1) hierarchisch aufgebaut und weisen eine Baumstruktur auf. Als Knoten des Baumes können auftreten:

- Elemente – in Form von Start- und Endtag (`<tag-name>`,`</tag-name>`), bzw. Empty-Element-Tag (`<tag-name />`)
- Attribute – Zusatzinformation vorkommend als Wertepaar im Start- oder Empty-Element-Tag (`<tag-name attribut-name="attribut-wert" />`)
- Steueranweisungen – `<?name daten?>`
- Kommentare – `<!-- Kommentar -->`
- Text – repräsentiert als normaler Text oder als CDATA-Abschnitt (`<![CDATA[text]]>`)

CDATA ist ein Akronym für "Character Data". Man verwendet CDATA-Abschnitte, um gewisse Textbereiche vom Auswerten eines Parsers zu schützen. Dieser Fall tritt ein, wenn ein Textknoten Markups enthält, die nicht als solche ausgewertet werden sollen.

Es wird im Allgemeinen von der Wohlgeformtheit eines XML-Dokuments gesprochen, wenn das Dokument die Auszeichnungsregeln des W3C-Standards befolgt. Dazu gehören unter anderem:

- Eine gültige Deklaration von folgender, exemplarischer Form:
`<?xml version="1.0" encoding="UTF-8" standalone="yes"?>`
Das Attribut "version" muss als Wert die Nummer des verwendeten Auszeichnungsstandards beinhalten, als Wert der Variablen "encoding" wird der verwendete Zeichensatz angegeben und die Variable "standalone" gibt an, ob eine Dokumenttypdefinition vorliegt oder nicht.
- Elementnamen (Tagnamen) müssen mit einem Buchstaben, Doppelpunkt oder Unterstrich, dürfen aber nicht mit den Teilstrings "xml" oder "XML" beginnen, im weiteren Verlauf des Namens dürfen auch Zahlen, Bindestrich und Punkt verwendet werden.
- Attributnamen unterliegen der gleichen Konvention wie Elementnamen, Attributwerte müssen von Anführungszeichen umschlossen sein.
- Eine korrekte Hierarchie. Es muss genau ein Wurzelement im Hierarchiebaum des XML-Dokuments vorhanden sein. Desweiteren dürfen sich Elemente unterschiedlichen Typs nicht überlappen, Verschachtelungen hingegen sind legitim.

Für die vollständige Beschreibung des XML-Standards sei an dieser Stelle auf die XML-Spezifikation des W3C verwiesen. [14]

Folgendes Beispiel soll den prinzipiellen Aufbau eines XML-Dokuments verdeutlichen:

```
<?xml version="1.0" encoding="UTF-8"?>
<seminararbeit>
  <chapter number="1">
    <title>Einführung</title>
    <paragraph>Es war einmal...</paragraph>
  </chapter>
  <chapter number="2">
    <title>Die Geschichte des World Wide Web</title>
    <paragraph>Zu Beginn des...</paragraph>
    <paragraph>Die Fähigkeit...</paragraph>
  </chapter>
  <chapter number="3">
    <title>Die Technik des Web 2.0</title>
    <paragraph>In den folgenden...</paragraph>
    <chapter number="3.1">
      <title>(X)HTML und CSS</title>
      <paragraph>Der Aufbau von...</paragraph>
    </chapter>
    <chapter number="3.2">
      <title>JavaScript</title>
      <paragraph>Die erste Version...</paragraph>
    </chapter>
  </chapter>
</seminararbeit>
```

Listing 1: Beispiel eines XML-Dokuments

3.4.1 DTD

In Kapitel 3.4 wurde erläutert, welche Kriterien ein XML-Dokument erfüllen muss, um nach dem W3C-Standard "wohlgeformt" zu sein. Für einige Anwendungsgebiete ist die Wohlgeformtheit ein hinreichendes Kriterium. Soll ein XML-Dokument jedoch auch "gültig" sein, das heißt von einem Parser auf die Einhaltung bestimmter Regeln überprüft werden, so müssen eben genau diese Regeln aufgestellt und in den Kontext des XML-Dokuments gebracht werden. Dies geschieht mittels der "Document Type Definition" (zu deutsch Dokumenttypdefinition). Sie legt die Art des Inhalts von Attributen sowie die Reihenfolge und Verschachtelung der einzelnen Elemente fest. Bekannte Dateiformate wie SVG für Vektorgrafiken, MathML für mathematische Formeln und WML für die Darstellung von Inhalten auf Displays von Handys sind Dokumenttypdefinitionen für XML. Um das Konzept der DTD zu verdeutlichen, wird im Folgenden das Beispiel aus Kapitel 3.4 um eine Dokumenttypdefinition erweitert. Als Regeln kommen zum Einsatz:

- Eine Seminararbeit besteht aus einem oder mehreren Kapiteln
- Ein Kapitel hat immer genau eine Überschrift gefolgt von einem oder mehreren Absätzen und keinem bis beliebig vielen weiteren Kapiteln
- Jedes Kapitel muss ein Attribut "number" besitzen
- Überschriften und Absätze bestehen aus Zeichendaten

```
<!ELEMENT seminararbeit (chapter)+>
<!ELEMENT chapter (title, paragraph+, chapter*)>
<!ATTLIST chapter number CDATA #REQUIRED>
<!ELEMENT title (#PCDATA)>
<!ELEMENT paragraph (#PCDATA)>
```

Listing 2: Beispiel einer DTD

Wie in Kapitel 3.4 schon erwähnt, ist das Schlüsselwort (P)CDATA ein Akronym für "(parsed) character data" und steht für eine beliebige Folge von Zeichendaten. Im Unterschied zu PCDATA wird der Text in CDATA nicht durch den Parser geparkt, eventuell auftretende Tags innerhalb des Textes werden als reiner Text interpretiert. Wird die obige DTD als externe Datei in Form von `<!DOCTYPE seminararbeit SYSTEM "seminararbeit.dtd">` in das XML-Beispiel aus Kapitel 3.4 integriert, kann ein Parser das Dokument verifizieren und, falls kein Fehler vorliegt, als "gültig" akzeptieren. [1, 14]

Als Weiterentwicklung von Dokumenttypdefinitionen ist "XML Schema", eine Empfehlung des W3C, zu nennen. Im Unterschied zu DTDs wird die Struktur hier in Form eines XML-Dokuments beschrieben. Da XML Schemata und DTDs prinzipiell die gleiche Aufgabe erfüllen, nämlich ein Dokument auf bestimmte Regeln hin überprüfen zu können, XML Schemata aber weitaus komplexer als DTDs sind, wird im Umfang dieser Arbeit nicht näher auf XML Schema eingegangen. [20]

3.4.2 XSLT

Da XML bis jetzt nicht, wie von den Entwicklern erwartet oder gehofft, die Auszeichnungssprache für das WWW ist, es sich aber als allgemeine Auszeichnungssprache für Dokumente mehr und mehr durchsetzt, muss ein Bogen zwischen der strukturierten Datenhaltung im XML-Format und der Präsentation der Daten im Web als HTML-Dokument geschlagen werden. Hier setzt XSLT (Extensible Stylesheet Language Transformation) an. Mit Hilfe von XSLT ist es möglich, XML-konforme Dokumente in Dokumente mit anderer Struktur zu transferieren. Herauskommen können dabei auch durchaus Dokumente, die nicht dem XML-Standard genügen. Da XML wie auch z.B. HTML-Dokumente Baumstrukturen aufweisen, spricht man von der Transformation eines Quellbaums in einen Ergebnisbaum. Zu diesem Zweck werden die in einer XSL-Datei definierten Muster gegen Elemente des Quellbaums getestet. Wird ein Muster im Quellbaum gefunden, so wird ein Teil des Ergebnisbaums durch Instanzieren eines ebenfalls in der XSL-Datei definierten Templates generiert.

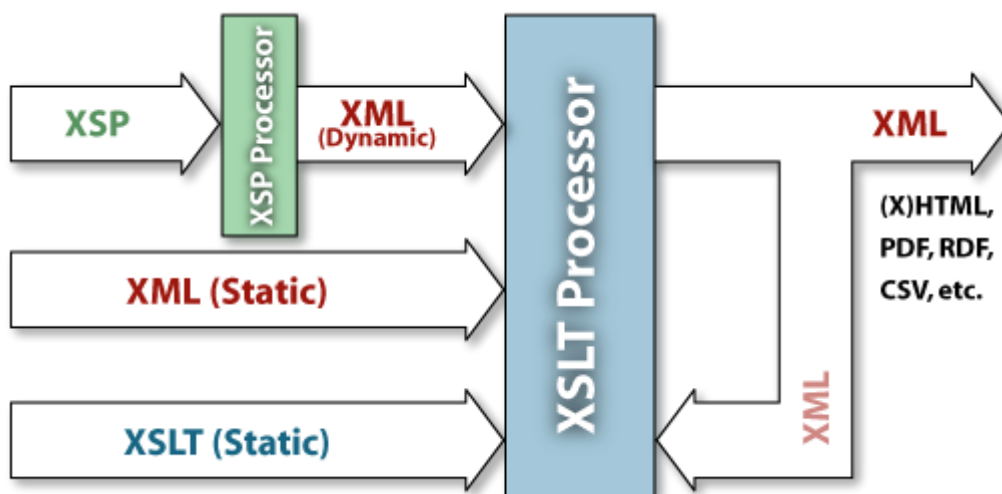


Abbildung 4: Funktionsweise von XSLT

Zur Verdeutlichung bekommt das Beispiel aus Kapitel 3.4 eine XSL-Transformationsdatei, um das Dokument grafisch aufbereitet in einem Browser darstellen zu können. Durch den folgenden Beispielcode wird im XSLT-Prozessor des Browsers eine komplette HTML-Datei generiert. [1, 29]

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html><head></head><body style="font-family:Verdana; color:black">
      <xsl:apply-templates />
    </body></html>
  </xsl:template>
  <xsl:template match="chapter">
    <br />
    <h3>
      <xsl:value-of select="@number" />
      <xsl:text> </xsl:text>
      <xsl:value-of select="title" />
    </h3>
    <xsl:apply-templates />
  </xsl:template>
  <xsl:template match="title">
    <span style="display:none"><xsl:value-of select="." /></span>
  </xsl:template>
  <xsl:template match="paragraph">
    <p><xsl:value-of select="." /></p>
  </xsl:template>
</xsl:stylesheet>

```

Listing 3: Beispiel eines XSL-Dokuments

Durch die "xsl:template"-Elemente werden Muster im Quellbaum gesucht, die dem Wert des Attributes "match" entsprechen und durch den Inhalt der Template-Elemente ersetzt. Das "xsl:apply-templates"-Element veranlasst die weitere Ersetzung von Mustern durch Templates innerhalb eines Templates. Auf diese Weise können hierarchische Strukturen verarbeitet werden. Mittels "xsl:value-of" kann der Wert eines XML-Objekts oder der Wert eines seiner Attribute in den Ergebnisbaum transferiert werden.

Die Ausgabe des Browsers sieht folgendermaßen aus:

<p>1 Einführung</p> <p>Es war einmal...</p> <p>2 Die Geschichte des World Wide Web</p> <p>Zu Beginn des...</p> <p>Die Fähigkeit...</p> <p>3 Die Technik des Web 2.0</p> <p>In den folgenden...</p> <p>3.1 (X)HTML und CSS</p> <p>Der Aufbau von...</p> <p>3.2 JavaScript</p> <p>Die erste Version...</p>

Listing 4: Ausgabe der Beispiellistings

Neben den im Beispiel verwendeten Template-Regeln verfügt XSLT über Schleifen und bedingte Verarbeitung, Sortieralgorithmen, Variablen und vielen weiteren Funktionen, auf die im Umfang dieser Arbeit nicht weiter eingegangen wird. [1]

3.4.3 XMLHttpRequest

Das XMLHttpRequest Objekt wurde erstmals von Microsoft als ActiveX-Objekt im Internet Explorer 5 implementiert. Mittlerweile ist es in allen "großen" Browsern als natives Objekt verfügbar. Das XMLHttpRequest-Objekt ermöglicht es einer Skriptsprache HTTP-Anfragen an einen Webserver zu starten ohne die Webseite neu laden zu müssen. Somit ist es möglich, HTTP-Anfragen und deren Antworten vom Server im Hintergrund abzuwickeln ohne dass der Anwender dies wahrnimmt. Diese Eigenschaft ermöglicht eine asynchrone Kommunikation und macht das XMLHttpRequest-Objekt zu einem wichtigen Teil der Ajax-Technik. [2, 17]

3.5 Ajax

Das Wort Ajax ist ein Apronym für "Asynchronous JavaScript And XML", einer Entwicklungstechnik für interaktive Webseiten. Wie sich aus dem Namen schon vermuten lässt, ist Ajax ein Zusammenschluss verschiedener bereits bestehender Techniken. Aus diesem Grund kann die Entwicklung keinem Unternehmen und keiner Organisation zugeordnet werden. Es ist jedoch offensichtlich, dass Google mit dem Gebrauch des XMLHttpRequest-Objektes in seinen Produkten wie z.B. "Google Maps" und "Google Suggest" maßgeblich an der Verbreitung von Ajax beteiligt ist. [2, 21]

Bei der Verwendung von Ajax kommen folgende Techniken zum Einsatz: [4]

(X)HTML und CSS	Erscheinungsbild
Document Object Model	dynamische Ausgabe
XML und XSLT	Datenaustausch und -manipulation
XMLHttpRequest	asynchrone Datenanforderung
JavaScript	Bindeglied

Tabelle 3: Ajax Technik und ihre Aufgaben

Die Motivation hinter der Entwicklung und Verwendung von Ajax ist es, Web-Anwendungen intuitiver und interaktiver zu gestalten, um eine Alternative zu Desktop-Anwendungen darzustellen. Der Unterschied zu dem schon seit 10 Jahren existierenden DHTML ist einen Bruch des Request-Response-Paradigmas (s. Abb. 5). Es wird also nicht mehr bei jeder Benutzeraktion ein Request an den Webserver geschickt, der daraufhin mit einem Response, meist einer kompletten HTML-Seite, antwortet. Vielmehr interagiert der Anwender nur noch mit der AJAX-Engine, die clientseitig Aufgaben erledigt und nur bei Bedarf – auch ohne direkte Aktion des Anwenders – Anfragen an den Server schickt. Die Antworten des Servers sind in der Regel keine kompletten HTML-Seiten, sondern nur noch kleine XML-Teile, die mittels JavaScript asynchron in das DOM der bestehenden Webseite integriert werden. Die Folgen dieser asynchronen Kommunikation machen sich für den Benutzer in schnelleren Antwortzeiten sowie intuitiverer Bedienung und für die Betreiber der Web-Dienstleistung in sinkender Serverauslastung positiv bemerkbar. [2, 21]

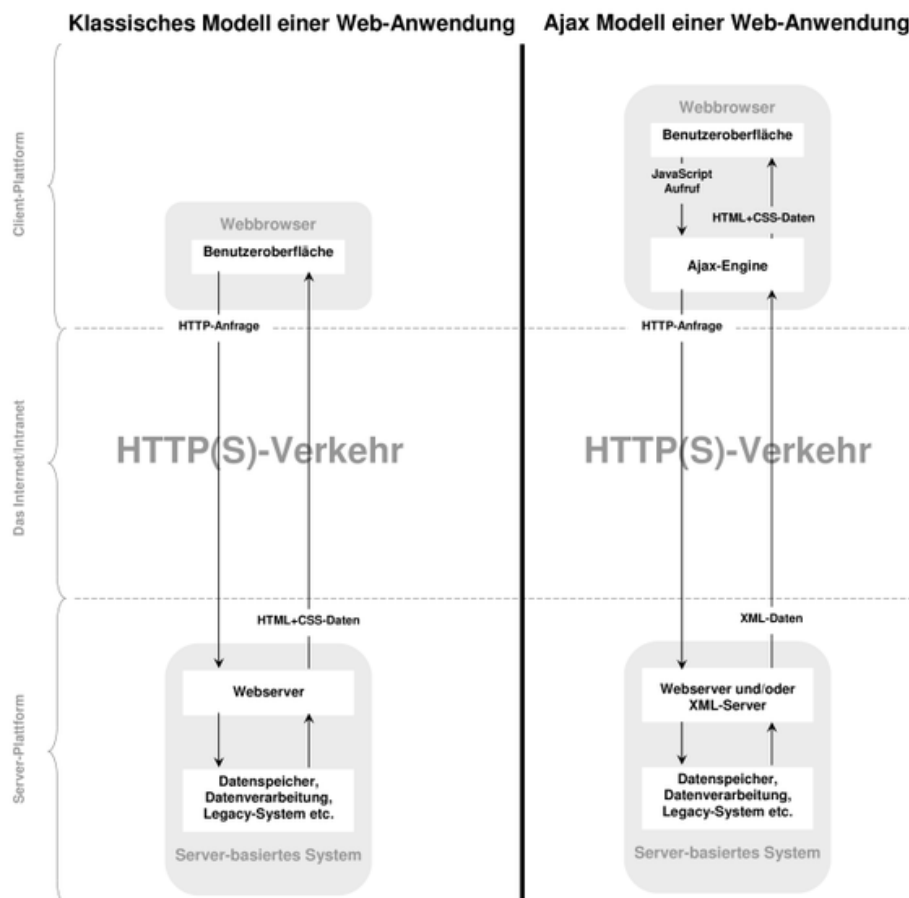


Abbildung 5: Kommunikationsmodelle

3.6 Web Services

Web Services sind Software-Anwendungen, die – im Gegensatz zur üblichen Client/Server-Architektur – über bestimmte Schnittstellen direkte Interaktion mit anderen Software-Agenten durchführen. Beim Client/Server-Ansatz greift der Client auf die Daten und Funktionen des Servers zu und stellt die ihm von Server übermittelten Inhalte im Browser dar. Bei Web Services hingegen werden Daten und Funktionalitäten zwischen Anwendungen ausgetauscht. Dies geschieht meist mittels XML-Nachrichten, die über Internet-Protokolle, meist HTTP, versandt und empfangen werden. [5, 27]

Wie viele andere Techniken bestehen Web Services aus einer Vielzahl verschiedener Standards und Spezifikationen. Drei Elemente werden als Grundlage von Web Services angesehen:

- **Übermittlung:** Nachrichten müssen zwischen Service-Anbieter, Service-Konsument und Service-Verzeichnis übermittelt werden.
- **Beschreibung:** Damit der Service-Konsument weiß was ihm geboten wird, muss der Service und die von ihm zur Verfügung gestellten Methoden beschrieben werden.
- **Verzeichnisdienst:** Services sollten in ein Service-Verzeichnis aufgenommen und dort gefunden werden können.

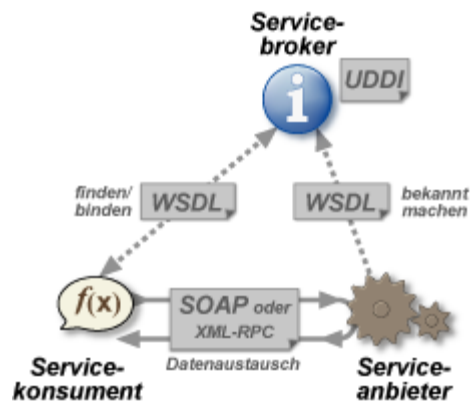


Abbildung 6: Die Service-orientierte Architektur

Als Technik zur Nachrichtenübermittlung hat sich SOAP gegenüber der konkurrierenden Technik XML-RPC durchgesetzt. SOAP basiert auf XML und wird vom World Wide Web Consortium verwaltet. Ursprünglich war SOAP ein Acronym für "Simple Object Access Protocol". Da es aber weder simpel ist, noch dem Objektzugriff dient, steht SOAP ab der Version 1.2 nur noch für sich selbst. [5, 16]

Für die Beschreibung für Web Services hat sich WSDL (Web Services Description Language) als Standard durchgesetzt. Obwohl WSDL oft als grundlegender Standard für Web Services genannt wird, ist SOAP eigentlich ausreichend. Die Beschreibung mittels WSDL kann als eine Art Schnittstelle zwischen dem Client und dem Webservice-Anbieter verstanden werden. Erst liest der Client per WSDL die zur Verfügung gestellten Services und greift dann über SOAP darauf zu. [5, 18]

Als Standard für den Verzeichnisdienst hat sich UDDI (Universal Description, Discovery and Integration of Web Services) durchgesetzt. Im Ursprungsjahr der Web Services, 1999, entwickelte Microsoft ein proprietäres Verzeichnisdienstformat namens DISCO. Schnell wurde jedoch klar, dass die Funktionalität nicht ausreichend war, um Web Services-Verzeichnisse zu realisieren. Das UDDI-Projekt, welches am 6. September 2000 von den Firmen Ariba, IBM und Microsoft gegründet wurde sollte hier Abhilfe schaffen. Mittlerweile ist die 3. Version von UDDI durch das offizielle Standardisierungsgremium OASIS standardisiert. [5]

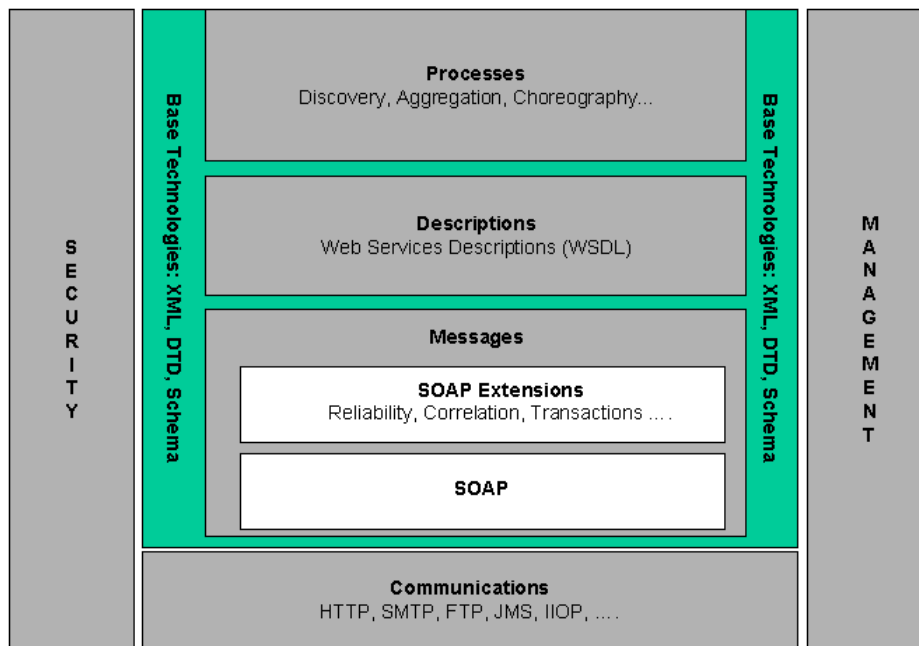


Abbildung 7: Web Services Architektur Stapel

Die wohl bekanntesten Beispiele für Web Services kommen von Google und Amazon. Mit Googles Web Service ist es möglich, die Google-Suche auf private Webseiten zu nutzen und Amazon bietet die Shop-Suche und das deutsche Partnerprogramm als Web Service an. So lädt beispielsweise das Musikprogramm Amarok per Web Service alle Cover der in der Playlist vorhandenen Alben von den Amazon-Servern auf den lokalen Rechner.

3.7 Really Simple Syndication

RSS ist eine Technik zum abonnieren von Informationen, die auf Webseiten veröffentlicht werden. Zu diesem Zweck bietet der Anbieter – meist eine Nachrichtenseite oder ein Weblog – einen sogenannten Webfeed (auch RSS-Feed oder RSS-Channel genannt) in Form einer XML-Datei an. Der Abonnent kann sich diese Datei herunterladen oder die Informationen direkt in eigene Webseiten integrieren. Um die Informationen unterschiedlicher RSS-Feeds zu sammeln und strukturiert darzustellen wurden sogenannte Aggregatorprogramme oder Feed-Reader entwickelt. Die Funktionen dieser Aggregatorprogramme werden mittlerweile auch in E-Mail Programme oder Webbrowser integriert. Der Anwender hat gegenüber der herkömmlichen Informationsgewinnung den Vorteil, dass die Informationen nicht durch zeitaufwendiges besuchen unterschiedlichster Webseiten gesammelt werden müssen, sondern durch ein zentrales Programm abgerufen und auch archiviert werden können.



Abbildung 8: Ein verbreitetes Feed-Icon

Auch der Name RSS ist ein Akronym, er ist jedoch in den verschiedenen technischen Spezifikationen unterschiedlich definiert. Dies lässt sich vor allem darauf zurückführen, dass verschiedene Entwicklergruppen teilweise unabhängig voneinander unterschiedliche Versionen von RSS herausgaben. Den Anfang machte Netscape im Jahr 1999 mit der Version 0.90, basierend auf RDF. RSS stand bei dieser Version für **"RDF Site Summary"**. Nur vier Monate später veröffentlichte Dan Libby, der Entwickler der ersten Version die Version 0.91, welche nicht mehr auf RDF sondern auf einer Dokumenttypdefinition für

XML basierte und das Format vereinfachte. Nun wurde im Zusammenhang mit RSS von "Rich Site Summary" gesprochen. Trotz der aufkommenden Verbreitung von RSS verlor Netscape das Interesse an dem Format und es bildeten sich zwei unterschiedliche Entwicklungszweige. Der amerikanische Softwarehersteller UserLand entwickelte die Version 0.9x weiter und veröffentlichte im Jahr 2000 seinerseits eine Version 0.91, die nicht mit der vorangegangenen Spezifikation übereinstimmt. Es folgte die Version 0.92 und Entwürfe für die Versionen 0.93 und 0.94. Ebenfalls im Jahr 2000 veröffentlichte eine unabhängige Entwicklergruppe RSS 1.0. Diese Version basiert erneut auf RDF. Im Jahr 2002 veröffentlichte UserLand ihre Weiterentwicklung der 0.9x-Reihe mit der Version 2.0. Diese Version wird als "Really Simple Syndication" bezeichnet. [9, 10, 24, 25]

Aus dem Bedürfnis heraus, die Vorteile aus den verschiedenen RSS-Versionen mit zusätzlichen Weiterentwicklungen zu einem neuen Standard zu vereinen, entstand 2003 ein neues Format, welches nach mehreren Namenswechslern Atom genannt wurde. Schon in der Version 0.3 trieb Google mit seinen Produkten die Verbreitung des Standards voran, die Version 1.0 aus 2004 wurde im Dezember 2005 zu einem offiziellen Internetstandard. [22]

4 Zusammenfassung

Das World Wide Web ist ein komplexes und langlebiges Medium. Seit Entstehung der ersten Webseiten sind nun über zehn Jahre vergangen, in denen immer neuere und meist auch bessere Standards verabschiedet und Techniken entwickelt wurden. Die Nutzeranzahl des Webs steigt ebenso wie die Bandbreite der Internetzugänge. Die aufgeführten Techniken tragen zur Struktur, Ästhetik und Benutzerfreundlichkeit des Netzes wie wir es heute kennen bei, ein Ende der Entwicklung ist jedoch nicht abzusehen. Das WWW als Universalplattform für Anwendungen aller Art zu betrachten mag etwas futuristisch klingen, jedoch sollte man sein Potenzial nicht unterschätzen.

"Nobody will ever need more than 640k RAM!"

sagte Bill Gates 1981. Sollte es im Blick auf diese Aussage nicht auch erlaubt sein vom "Netz als Betriebssystem" zu träumen?

5 Literatur

- [1] Behme, Henning; Mintert, Stefan: XML in der Praxis. [http://www.linkwerk.com/pub/xmlidp/2000, 2000](http://www.linkwerk.com/pub/xmlidp/2000,2000)
- [2] Bergmann, Olaf; Bormann, Michael: AJAX – Frische Ansätze für das Web-Design, Bremen, 2005
- [3] Bosak, Jon; Bray, Tim: XML and the Second-Generation Web. Scientific American Magazine, Mai 1999
- [4] Garrett, Jesse James: Ajax: A New Approach to Web Applications. <http://www.adaptivepath.com/publications/essays/archives/000385print.php>, 2005
- [5] Hauser, Tobias; Löwer, Ulrich: Web Services, 2004
- [6] Kappel, Gerti; Kramler, Gerhard; Retschitzegger, Werner: XML – Grundlagen, http://www.big.tuwien.ac.at/teaching/offer/ss02/uml_vo/XML_Grundlagen_screen.pdf, Wien, 2002
- [7] McLellan, Drew: Very Dynamic Web Interfaces, <http://www.xml.com/pub/a/2005/02/09/xml-http-request.html>, 2005
- [8] O'Reilly, Tim: What Is Web 2.0. <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>, 2005
- [9] Pilgrim, Marc: What is RSS. <http://www.xml.com/pub/a/2002/12/18/dive-into-xml.html>, 2002
- [10] RSS-DEV Working Group: RDF Site Summary (RSS) 1.0. <http://web.resource.org/rss/1.0/spec>, 2000
- [11] Seeboerger-Weichselbaum, Michael; JavaScript – Das bhv Taschenbuch, 2000
- [12] SELFHTML e.V.: SelfHTML 8.1.1 – HTML-Dateien selbst erstellen, <http://de.selfhtml.org>, 2005
- [13] W3C: Document Object Model (DOM) Level 1 Specification. <http://www.w3.org/TR/REC-DOM-Level-1>, 1998
- [14] W3C: Extensible Markup Language (XML) 1.0 (Third Edition). <http://www.w3.org/TR/REC-xml>, 2004

- [15] W3C: HTML 4.01 Specification. <http://www.w3.org/TR/html4>, 1999
- [16] W3C: SOAP Version 1.2 Part 1: Messaging Framework, <http://www.w3.org/TR/soap12-part1>, 2003
- [17] W3C: The XMLHttpRequest Object. <http://www.w3.org/TR/XMLHttpRequest>, 2006
- [18] W3C: Web Services Description Language (WSDL) 1.1, <http://www.w3.org/TR/wsdl>, 2001
- [19] W3C: XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition). <http://www.w3.org/TR/xhtml1>, 2002
- [20] W3C: XML Schema, <http://www.w3.org/XML/Schema>, 2004
- [21] Wikipedia: Ajax (Programmierung). http://de.wikipedia.org/wiki/Ajax_%28Programmierung%29, Stand 05.05.2006
- [22] Wikipedia: Atom (standard). http://en.wikipedia.org/wiki/Atom_%28standard%29, Stand 12.05.2006
- [23] Wikipedia: Document Object Model. http://de.wikipedia.org/wiki/Document_Object_Model, Stand 05.05.2006
- [24] Wikipedia: RSS. <http://de.wikipedia.org/wiki/RSS>, Stand 12.05.2006
- [25] Wikipedia: RSS (file format). http://en.wikipedia.org/wiki/RSS_%28file_format%29, Stand 12.05.2006
- [26] Wikipedia: Web 2.0. http://de.wikipedia.org/wiki/Web_2.0, Stand 22.06.2006
- [27] Wikipedia: Web Service. http://de.wikipedia.org/wiki/Web_Service, Stand: 22.06.2006
- [28] Wikipedia: World Wide Web. http://de.wikipedia.org/wiki/World_Wide_Web, Stand 14.06.2006
- [29] Wikipedia: XSL Transformation, http://de.wikipedia.org/wiki/XSL_Transformation, Stand 22.05.2006
- [30] Zschunke, Peter: "Auf Wiedersehen, Word". <http://www.spiegel.de/netzwelt/technologie/0,1518,druck-414039,00.html>, 2006