

Das Web der Zukunft

– Tendenzen, Entwicklungen und Visionen

„Wer Visionen hat, sollte zum Arzt gehen.“

Altkanzler Helmut Schmidt

Seminar Kommunikationsnetze und -Protokolle im Sommersemester 2009

Verfasser: Christoph Nolte
Kursnummer: 21747
Datum des Seminars: 24.-25.09.2009

FernUniversität in Hagen

Inhaltsverzeichnis

Das Web der Zukunft	
– Tendenzen, Entwicklungen und Visionen.....	i
Abbildungsverzeichnis.....	ii
1. Über Prognosen.....	1
2. Vergangenheitsbetrachtungen als Basis einer Zukunftsschau.....	1
3. Aufgabenverteilung.....	3
4. Netzbasierte Dienste.....	3
4.1 Klassische Webapps.....	4
4.2 Client-basierte Webapps.....	5
4.3 Die Verschmelzung von Web applications und Desktop-Programmen.....	6
4.4 Der Desktop im Netz.....	6
5. Web-Standards.....	7
5.1 Der richtige Zeitpunkt.....	7
5.2 Das HTML der letzten beiden Dekaden.....	8
5.2.1 Inkompatible Browser.....	8
5.2.2 Plugins.....	8
5.3 HTML 5.....	9
5.3.1 Vorgeschriebenes Browserverhalten.....	9
5.3.2 Verbesserte Semantische Dokumentauszeichnung.....	9
5.3.3 Einbettung von Multimedia.....	9
5.3.4 Eingebettetes SVG und MathML.....	10
5.3.5 Erweiterte Unterstützung von Web-Applikationen.....	10
6. Mobile Geräte.....	10
7. Das semantische Web.....	11
7.1 Wie Semantik unser Leben erleichtern könnte.....	11
7.2 Verschlagwortung von Daten.....	12
7.3 Verknüpfung von Daten.....	13
7.4 Technische Umsetzung des semantischen Webs.....	14
7.4.1 Mikroformate.....	14
7.4.2 RDFa.....	15
7.4.3 Upper case semantics.....	16
8. P2P-Technologie.....	16
8.1 Offene Entwicklung.....	16
8.2 Drahtlose Mesh-Netzwerke.....	17
8.3 Mögliche Realisierungen offener Hardware.....	17
9. Risiken.....	18
9.1 Überwachbarkeit.....	18
9.2 Informationsüberflutung.....	18
9.3 Spam.....	19
10. Fazit.....	20
Literaturverzeichnis.....	ii

Abbildungsverzeichnis

Abbildung 4.1: Prism ermöglicht Tray-Funktionen für Webapps [Prism]	6
Abbildung 5.1: Apokalypse der zwei Elefanten (Scan aus der englischen Buchausgabe).....	7
Abbildung 7.1: Links: Was der Computer sieht, rechts: Was der Mensch sieht. [W3C08].....	14

1. Über Prognosen

Glaubwürdige Zukunftsprognosen sind schwierig. Einer der am häufigsten begangenen Fehler dürfte die lineare Extrapolation sein. Als Beispiel mögen hier Aktienanalysten dienen: „Die Kurse sind schon seit einem halben Jahr gestiegen, also klare Kaufempfehlung“. Nach dem Crash können sie dann genau erklären, warum das ja jetzt so kommen musste und warnen vor einem Kauf, weil es ja „aus technischen Gründen“ NOCH weiter abwärts gehen müsse.

Im Bereich echter Wissenschaft und Technik begegnet einem dieser Denkfehler allerdings auch. So wirken heute etwa Science-Fiction-Visionen aus den Fünfziger Jahren des letzten Jahrhunderts niedlich, etwa wenn von dem Jahr 2000 die Rede ist, in dem wir in Mondkolonien wohnen und uns in schwebenden Autos mit Atomantrieb fortbewegen[Simmons07].

Entwicklungen scheitern aber nicht nur weil die physikalischen Gesetze für unsere Zwecke manchmal ungeeignet sind, sondern weil sie entweder völlig unwirtschaftlich sind oder einfach kein Bedarf vorhanden ist.

So dachten in den Achtzigern Produktmanager und Verkäufer sich wahrscheinlich ungefähr etwas wie „Früher hat jeder Radio gehört. Jetzt schauen alle lieber fern. Jetzt haben alle Telefon. Transferleistung: Alle wollen Bildtelefone von uns kaufen, das muss eine Goldgrube sein!“.

Obwohl man Bildtelefonie in vielen Post-Läden ausprobieren konnte, wollten nur die wenigsten selbst eins besitzen. Wahrscheinlich weil Nur-Sprache-Telefonie viele Vorteile hat. So macht es z.B. keinen Unterschied, ob man gerade aus dem Bett gefallen ist oder im Anzug mit Krawatte in einem ordentlich aufgeräumten Büro sitzt. Auch möchte man den Gesprächspartner vielleicht nicht ständig beim Reden ansehen müssen, sondern z.B. lieber nebenbei die Zeitung lesen.

Echte Innovationen tauchten oft relativ unangekündigt auf und ließen sich nicht einfach voraussehen. Bei Star Trek hat man schon früher kleine tragbare Kommunikationsgeräte benutzt, die heutigen Mobiltelefonen mit Klappdisplay nicht unähnlich waren. Niemand hätte sich damals aber gedacht, dass die wahre Cash-Cow für Mobilfunkbetreiber Teenager sind, die 20 Cent für 140-Zeichen-Textnachrichten ausgeben, welche über die denkbar unergonomischste Benutzerschnittstelle in dieses Ding mühsam eingetippt werden müssen.

Aber lassen wir uns überraschen... Diese Arbeit soll ein paar Ahnungen beschreiben, wie das Netz der Zukunft aussehen könnte.

2. Vergangenheitsbetrachtungen als Basis einer Zukunftsschau

Erst soll kurz die historische Entwicklung umrissen werden, um Gründe zu erkennen, warum die Menschheit überhaupt das Web in der heutigen Gestalt entwickelt hat. Nach der Frage, ob diese Ziele erreicht sind und welche Bedürfnisse mit weiter entwickelter Technologie besser gestillt werden könnten, werden verschiedene Ideen für die Zukunft beschrieben.

Schon immer gab es Gründe für Leute, Nachrichten auszutauschen. Das naheliegendste war, Kuriere mit der Depesche auf die Reise zu schicken, wenn es eilig war per Pferd oder als Brieftaube. Andere interessante Kommunikationsformen waren akustische Übertragung wie trommeln im Urwald oder jodeln in den Alpen, evtl. mit mehreren „Relay-Stationen“. Jede war der jeweiligen Umwelt gut angepasst und ging schneller als Briefe.

Das erste, was man als Kommunikationsnetz im weitesten Sinne bezeichnen konnte, waren mechanische Winkedinger, die erst in Frankreich, dann in ganz Europa eingesetzt wurden. Dies nannte man *optische Telegrafie*[WP01], die Datenrate war für heutige Verhältnisse relativ niedrig aber für die Kommunikationsbedürfnisse Napoleons wie z.B. Informationen über Feinde an den Landesgrenzen durchaus brauchbar. Man benötigte pro Station zwei Leute, einen mit Fernglas und einen anderen, der die Signalgeber ausrichtete.

Später umspannten Telegraphendrähte Europa, Russland und die USA. Bedient wurden die elektrischen

Telegraphenstationen wieder von Spezialisten, aber gegen entsprechende Gebühren konnten durchaus auch Privat- und Geschäftsleute davon Gebrauch machen.

Die Steigerung stellte Telefonie dar, die später in vielen Haushalten selbstverständlich wurde und somit nicht nur für Regierungen, Geschäftsleute oder Notfälle, sondern für breite Bevölkerungsschichten im Alltag zur Verfügung stand.

Während diese Kommunikationsnetze das Kommunikationsbedürfnis befriedigten, entstanden andere Formen zur Informationsbeschaffung und Unterhaltung. Namentlich waren diese Medien Radio und Fernsehen.

Die Technologietreiber Kommunikation, Information und Unterhaltung, sowie kommerzielle Ausbeutung sind für weitere Entwicklungen bedeutend.

Durch Miniaturisierung, Fortschritte in Computertechnik und Mikroelektronik wurden analoge Übertragungsformen abgelöst durch digitale.

Zur Informationsbeschaffung entwickelte die Deutsche Bundespost BTX, Frankreich war auf dem Gebiet mit einem ähnlichem System namens Minitel bereits sehr viel weiter. Nachdem die Telefonnetze nun digital sind, steht nach diversen Zwischenstufen wie Fido-Net, Mailboxen etc. nun auch jedermann heute der Zugang zum Internet offen.

Hatten anfangs nur Unis Zugang zu Usenet und Internet mit E-Mail, entspricht dies heute dem Standard für breite Schichten der Industrieländer und die ursprünglichen Kommunikationsbedürfnisse werden immer mehr darauf übertragen in Form von IP-Telefonie, Radio- und Video-Streams.

Das Internet erlebte eine rasante Entwicklung, welche hier nur knapp umrissen werden soll, damit Begriffe wie Internet und Web in einem Kontext eingebettet sind.

Nachdem das ARPA-Net einigermaßen funktionierte, wurden immer mehr Unis daran angeschlossen. 1989 erfand Tim Berners-Lee das WWW (world wide web) und mit Mosaik konnte man ein verteiltes System von verhyperlinkten Textdokumenten betrachten. In den Neunzigern wurde Internet auch für Privatleute erschwinglich und technisch überhaupt verfügbar.

Web bestand aus einfachen HTML-Seiten, die man sich mit dem vim selbst zusammenstrickte und die von „Surfern“ einfach per Browser abgerufen wurden. Moderne Firmen hatten auf einmal eine „Homepage“, wohinter sich im Prinzip eine Visitenkarte mit Bild und e-Mail-Adresse verbarg.

Browser waren beliebt, wenn sie möglichst viele Tags für Effekte unterstützte, wie z.B. <blink>. Es galt in erleuchteteren Kreisen als schlechter Stil, Frames zu benutzen und als gut, für Layout-Zwecke blinde (unsichtbare) Tabellen zu verwenden. Die Trennung von Form und Inhalt war also noch nicht verbreitet, solange man sich für einigermaßen gestalterisch anspruchsvoll hielt.

Sun versuchte mit Java im Browser plattformunabhängig interaktive Elemente zu ermöglichen, allerdings wurden Java-Applets oft als langsam empfunden und als einer der häufigsten Auslöser für Browserabstürze.

Viele User waren mit 33k6- oder 56k-Modems mit Ihrem Netzanbieter verbunden und bezahlten minutenweise. WWW wurde als WeltWeites Warten übersetzt und Inhalte (Texte und Bilder) sollten möglichst klein und effizient sein.

Die Welt dreht sich weiter und jetzt spielen Bandbreiten anscheinend keine Rolle mehr. Das Netz sieht ganz anders aus.

Inzwischen ist ADSL mit Pauschaltarif gängig, sodass es normal ist, sich große Videos und Live-Streams per Internet anzuschauen.

Java spielt Client-seitig eigentlich keine Rolle mehr, dafür ist es allerdings auf Servern eine recht populärere Technologie.

Zu HTML 4.01 und XHTML 1.0 existieren inzwischen brauchbare Implementierungen und mit CSS sowie DOM können interessante Möglichkeiten umgesetzt werden ohne die Standards proprietär „erweitern“ bzw. verletzen zu müssen. Techniken wie AJAX und das proprietäre Adobe Flash sind derzeit allgegenwärtig.

Die Bedeutung von Websites hat sich deutlich verschoben von Visitenkarten und Katalogen hin zu interaktiven Plattformen. Diese Tendenzen werden in den folgenden Kapiteln thematisiert.

3. Aufgabenverteilung

Früher bedeutete Computer ein Zimmer-großer Mainframe-Rechner mit einigen dummen Terminals. Die Arbeit in diesen verteilten Systemen teilte sich auf die Berechnungen im Großrechner auf und die bloße Anzeige bei den Bedienstellen.

Mit der Einführung von Workstations und PCs verschoben sich die Verhältnisse derart, dass die Clients in der Regel mit viel Rechenleistung und Arbeitsspeicher ausgestattet sind. Sie können auch komplexe Berechnungen selbst bewältigen und der Server wird entweder nicht benötigt oder dient bloß als zentraler Ort zur Datenablage.

Topologisch könnte man den historischen Zustand als fat-server/thin-client bezeichnen, woraufhin die clients allmählich immer „fetter“ wurden und die Server „abnahmen“.

Der Grund waren vor allem wirtschaftliche Erwägungen; während früher CPU und Arbeitsspeicher die teuersten Komponenten waren, versuchte man diese auf so viele Anwender wie möglich zu verteilen um sie möglichst effizient zu nutzen.

Nach dem Moor'schen Gesetz wurden Rechenleistung und Speicher immer billiger und „starke“ Einzelplatzrechner waren nicht mehr konzeptionell benachteiligt.

Heute sind Übertragungsbandbreiten ebenfalls billig und die Menschen möchten ihre Daten unabhängig von einem speziellen Rechner nutzen. Beispiele sind Adressinformationen, die sowohl im Kontaktprogramm auf dem PC vorhanden sein sollen, als auch auf dem PDA und Mobiltelefon. Hatte man früher einen Computer, sind heute für viele PC, Laptop, Netbook, PDA und Smartphone nicht unüblich.

Um wichtige Datenbestände synchron zu halten, setzen sich daher netzbasierte Dienste immer mehr durch, z.B. werden klassische MUAs¹ (mutt, pine, kmail, Mozilla Thunderbird etc.) abgelöst durch Web-basierte E-Mail-Programme.

Diese Programme lassen sich von allen Web-tauglichen Geräten gleichermaßen verwenden, welche mehr oder weniger zu bloßen Anzeigeegeräten mutieren, während die eigentliche Funktionalität wieder auf dem Server beheimatet ist.

Damit verschiebt sich aktuell der Trend wieder auf Fat-Server/thin-client. Aktuelle Technologien wie Apple iPhone und Palm WebOS sind ohne Web-basierte Dienste in ihren Funktionen stark beschnitten und praktisch nutzlos.

Dieser Trend vom individuellen Einzelplatzrechner hin zu netzbasierten Arbeitsweisen wird sich meiner Meinung nach fortsetzen. Wie und warum soll nachfolgend erläutert werden.

4. Netzbasierte Dienste

Software wird künftig weniger lokal installiert, sondern online bereitgestellt. Die klassischen Vorteile des thin-client-Modells bleiben; etwa ist solche Software zentral wartbar. Sie muss nicht auf tausenden Rechnern einzeln installiert werden, wobei auf vielen Probleme auftreten aufgrund exotischer Hardware, veralteter Software-Versionen oder anderen Inkompatibilitäten.

Ein anderer positiver Aspekt wäre eine hohe Datensicherheit aufgrund professioneller, regelmäßiger Backups und guter Behandlung der Hardware in klimatisierten Rechenzentren.

Softwareunternehmen gefällt die Idee, Programme nicht mehr zu verkaufen, sondern vermieten zu können². Damit ist ein beständigerer Einnahmestrom verbunden ist als mit einer CDROM, die einmal verkauft wird. Neues Geld ist höchstens dadurch zu erwarten, dass User (vielleicht und vielleicht auch erst nach ein paar

1 MUA: Mail User Agent

2 Buzzword: SaaS – Software as a Service

Jahren) eine neuere Version erwerben.

Weitere Vorteile für die Hersteller: Kein Ärger mehr mit „Raubkopierern“, Fehler können unbemerkt ausgebessert werden, Leistungen können nachträglich wieder entzogen werden. Ein aktuelles Beispiel ist Amazons E-Book-Reader „Kindle“, der gekaufte Bücher nachträglich wieder löschen kann[./09].

Fraglich ist natürlich, ob Anwender dieses neue Geschäftsmodell akzeptieren werden oder sich nicht lieber nach kundenfreundlicheren Alternativen umsehen. Im Bereich von Onlinespielen funktioniert dieses Konzept allerdings sehr gut und könnte in andere Bereiche überschwappen.

Auch sollte es bedenklich stimmen, wenn es darum geht, vertrauliche Daten einer (dubiosen? Wodurch hat sie mein Vertrauen verdient?) Firma anzuvertrauen und die eigene Kontrolle darüber aus der Hand geben. Stichworte wie „Bundestrojaner“ und „Datenpanne“ sind inzwischen auch außerhalb des informierten Fachpublikums bekannt. Der Gegenbeweis, nämlich dass vielen Leuten ihre persönlichen Daten völlig egal sind, ist mit einem Wort erbracht: StudiVZ. Allerdings könnte in der Zukunft ein gesellschaftlicher Lernprozess stattfinden, in der Leute sich richtig verhalten in einer Welt mit einem Gedächtnis, das nicht vergisst.

Wenden wir uns aber nun von der sozialen zur technischen Seite der netzbasierten Dienste.

Man kann unter dem Begriff *netzbasierte Dienste* verschiedenes verstehen, z.B. auch einen DNS-Server oder ein Desktop-Programm, welches als Jabber-Client dient. In dieser Betrachtung möchte ich mich damit jedoch auf Anwendungsprogramme beschränken, die in einem Webbrowser ablaufen. Das sind sogenannte *web applications* oder kurz *Webapps*.

In den Begriffen des vorherigen Kapitels ausgedrückt, sind Webapps meistens dem Thin-Client-Schema zuzuordnen, allerdings gibt es auch Tendenzen dahin, sie fetter werden zu lassen bzw. „smarter“, weil Intelligenz für Marketing-Leuten besser klingt als Übergewicht.

Und selbstverständlich ist es auch nichts neues, Programme per Netzwerk zu benutzen. Unix-User riefen ihre E-Mails schon vor zwanzig Jahren via telnet und mailx ab. Die Neuerung besteht in der „allgemein verständlichen“ Schnittstelle Webbrowser.

4.1 Klassische Webapps

Schon HTML 1.0 erlaubte es, Formulare zu erzeugen. Diese Formulare konnten Widgets zur Texteingabe, sowie radio buttons, Schaltflächen und ein paar weitere benutzen, wie man sie auch von Desktop-Programmen her kennt. Sinn von Formularen ist es, Daten an ein Programm auf dem Server zu senden. Dies geschieht durch eine standardisierte Schnittstelle – CGI³. Dieses Programm kann in jeder geeigneten Sprache geschrieben sein, häufig sind dabei PHP, Perl, Java und Ruby.

Klassische Suchmaschinen waren vermutlich die ersten Webapps. Der Anwender gibt einen Suchbegriff ein und kurz darauf wird in den Browser eine neue Seite geladen als Antwort.

CMS⁴, Forensoftware, Wikis und Weblogs mit Kommentarfunktion und Reservierungssysteme sind weitere Vertreter dieser Kategorie.

Eine wichtige Neuerung stellt AJAX⁵ dar. Diese Technik ermöglicht es, nur einzelne Teile einer Website nachzuladen, was viele Anwendungen erst ermöglicht. Ein gutes Beispiel ist Google Maps. Nach dem klassischen Verfahren hätte man vom Server das Bild einer Landkarte angezeigt bekommen mit ein paar Schaltflächen zur Navigation. Nach einem Klick auf die „Links“-Schaltfläche würde die Seite neu laden und anstatt Deutschland jetzt Frankreich in dem Bild zu sehen sein. Durch AJAX kann jetzt mit der Maus ein Kartenausschnitt flüssig bewegt und gezoomt werden. Fehlende Kartenteile werden bei Bedarf hinzugeladen. Im Prinzip ist AJAX eine Technologie, statische HTML-Seiten veränderbar zu machen. Im Gegensatz zu diversen vorherigen (Herstellerabhängigen) Versuchen wie Dynamic HTML (DHTML) hat sie sich bisher sehr

3 CGI: Common Gateway Interface

4 CMS: Content Management System

5 AJAX: asynchronous JavaScript and XML

gut bewährt.

Vor allem die Firma Google hat in den letzten Jahren Web-Basierte Programme stark weiterentwickelt, so wurden aus den einfachen Webmail-Diensten, welche es auch schon vorher gab, mit gmail eine ernstzunehmende Alternative zu MUAs.

Weitere Anwendungsfelder, die nicht ihren Ursprung im Internet haben wie E-Mail sind Kalenderprogramme und solche zur Textverarbeitung und Tabellenkalkulation.

An den Google-Varianten sind bereits einige Vorteile eines netzbasierten Ansatzes sichtbar. Ein interessantes Beispiel ist Funktion GoogleLookup() in dieser Büro-Suite. So kann etwa in der Textverarbeitungssoftware *Writely* die Formel `=GoogleLookup("Paraguay"; "internet users") [Google]` eingefügt werden und sichtbar ist die Anzahl der Internetnutzer in Paraguay. Weitere Beispiele sind:

- **„Countries and Territories** (like "Burkina Faso"): population, capital, largest city, gdp
- **U.S. States** (like "Tennessee"): area, governor, nickname, flower
- **Rivers** (like "Amazon River"): origin, length
- **Cities and Towns** (like "Chicago"): state, mayor, elevation
- **Musicians** (like "John Lennon"): date of birth, place of birth, nationality
- **Actors** (like "Audrey Hepburn"): date of birth, place of birth, nationality
- **Politicians** (like "Anwar Al-Sadat"): date of birth, place of birth, nationality
- **U.S. Presidents** (like "Zachary Taylor"): date of birth, place of birth, political party
- **Baseball Players** (like "Wade Boggs"): games, at bats, earned run average, position
- **Chemical Elements** (like "Helium"): atomic number, discovered by, atomic weight
- **Chemical Compounds** (like "Isopropyl Alcohol"): chemical formula, melting point, boiling point, density
- **Stars** (like "Betelgeuse"): constellation, distance, mass, temperature
- **Planets** (like "Saturn"): number of moons, length of day, distance from sun, atmosphere
- **Dinosaurs** (like "Velociraptor"): height, weight, when it lived
- **Ships** (like "USS Chesapeake"): length, displacement, complement, commissioned
- **Companies** (like "Hewlett-Packard"): employees, ceo, ticker“ [Google]

Im Prinzip gibt es momentan nur wenige Szenarien, in denen ein derartiges Vorgehen vorteilhaft gegenüber heutigen Verfahren erscheint. Es ist z.B. einfacher den Geburtstag von Barack Obama (4. August 1961) einmal irgendwo nachzuschlagen, als jedes Mal eine erneute Suche auszulösen wenn das Dokument geöffnet wird. Das trifft auf alle Daten zu, die sich wahrscheinlich nicht mehr ändern. Stark volatile Werte wie Börsenkurse oder Wetterdaten könnten hingegen von solch einer Vernetzung profitieren. Allerdings existiert die Möglichkeit, externe Daten in Dokumente einzubinden auch in Desktop-Bürosuiten. Man wird sehen, wie sich die Dinge entwickeln werden.

Ein weiteres Beispiel ist ein online nutzbares Bildbearbeitungsprogramm der Firma Adobe[heise].

Einer der wichtigsten Entwicklungstreiber im Computerbereich soll hier auch nicht unerwähnt bleiben – der Spieltrieb. Web-basierte Spiele können von vielen Teilnehmern gleichzeitig gespielt werden, was offensichtlich auch genutzt wird. So waren etwa in dem ersten großen deutschen Web-Spiel *Galaxywars* Ende 2004 knapp 200000 (!) Mitspieler registriert[GW04]. Diese Art von Spielen sind oft Aufbau- und Wirtschaftssimulationen in Echtzeit. „Echtzeit“ ist in diesem Fall allerdings nicht gleichbedeutend mit Hektik, da es mitunter ein paar Tage dauert, bis ein Raumschiff sein Ziel erreicht hat und neue Befehle annehmen kann.

Andere interessante Technologien sind Dabble DB (<http://dabbledb.com/>), womit sehr einfach Datenbankanwendungen erzeugt werden können und Yahoo! Pipes (<http://pipes.yahoo.com/>), welches dazu dient, *Mashups* zu bilden. Auf diese Konzepte soll hier aber nicht weiter eingegangen werden.

4.2 Client-basierte Webapps

Eine HTML-Datei mit eingebettetem Javascript kann selbstverständliche auch lokal gespeichert und ohne

Netzanbindung sinnvolle Funktionen erbringen. Ein beeindruckendes Beispiel dafür ist das Programm TiddlyWiki, welches wirklich nur aus einer einzigen Webseite besteht, die man sich z.B. auf einen USB-Speicherstick ablegt (<http://www.tiddlywiki.com/>).

Ob Daten für Plugins wie die weit verbreiteten Flash-Spiele Webapps sind, ist umstritten, da es sich hierbei um keine offenen Web-Standards handelt. Eine Begründung dieser Ansicht folgt in Kapitel 5.2.2.

4.3 Die Verschmelzung von Web applications und Desktop-Programmen

Andere Entwicklungen zielen auf sogenannte *smart clients*, die sich am ehesten beschreiben lassen als thin clients mit Hardwarebeschleunigung und Zugriff auf lokale Dateien.

Kernidee ist, Webapps wie lokale Programme nutzen zu können.

Ein häufig benutzter Begriff dafür ist auch *RIA*: Rich Internet Application. Unter diesem Marketing-Begriff vermuten die üblichen Verdächtigen das große Geschäft der Zukunft[Rapoza08].

Adobe schnürt sein Flash mit Webkit und ein paar anderen Komponenten zusammen und nennt dieses Paket „AIR“ – *Adobe Integrated Runtime*. Microsoft versucht mit seinem „Silverlight“ Marktanteile zu sichern und die Google-Variante heißt „Google Gears“. Eine weitere Plattform nennt sich Curl. Diese ist mit 10 Jahren Entwicklung für Geschäftsanwendungen vermutlich die vernünftigste Wahl[Rapoza08].

Neben diesen proprietären Versuchen existiert für RIAs auch freie Software, am bekanntesten ist wohl „Prism“ von Mozilla. Prism hieß früher WebRunner und verfolgt einen anderen Ansatz als die oben angeführten Technologien. Während diese versuchen, den Browser irgendwie zu erweitern, sodass Webseiten sich wie Desktop-Programme verhalten, wird hier der technologischen Entwicklung der letzten Jahre Rechnung getragen, dass Websites mit AJAX u.Ä. dieses auch ohne Erweiterungen können.

„Mit Prism lassen sich Webapplikationen aus dem Browser herausbrechen und als eigenständige Applikationen nutzen. Prism basiert dabei auf Firefox, stellt die Applikationen aber in einem eigenen Fenster ohne die browsertypischen Bedienelemente dar.“[Ihlenfeld07] Dieses Konzept wird auch Site Specific Browsers (SSB)[MozWiki09] genannt.

Dabei ergeben sich folgende Vorteile gegenüber der Ausführung in normalen Browsern[Prism]:

Webapps können per Icon gestartet werden, auch automatisch bei jedem Login, und wie andere Programme auch per Taskleiste angewählt werden. Es besteht die Möglichkeit, die Seite in ein „Tray“ zu minimieren und die üblichen unaufdringlichen Benachrichtigungen zu nutzen, etwa wenn in Google Mail eine neuer E-Mail empfangen wurde (Abbildung 4.1).

Vorteilhaft ist auch die erhöhte Stabilität durch die Verwendung separater Prozesse für jede Prism-Instanz; stürzt Firefox ab, wird die Webapp davon nicht beeinflusst.

Ein Vorteil gegenüber den proprietären RIA-Frameworks ist, dass Prism keine Plattform ist, sondern ausschließlich auf offenen Webstandards basiert, inklusive HTML, JavaScript, CSS und <canvas>. An 3D-Hardware-Unterstützung und der Möglichkeit, Daten lokal persistieren zu können, wird gearbeitet[MozWiki08].



Abbildung 4.1: Prism ermöglicht Tray-Funktionen für Webapps [Prism]

Meine Prognose: In nicht allzu ferner Zukunft werden Webapps so gestaltet werden können, dass sie sich in ihrem Aussehen und Bedienverhalten nicht mehr von Desktop-basierten unterscheiden.

4.4 Der Desktop im Netz

„Verschmelzung“ von Desktop und Web funktioniert auch in die andere Richtung: Anstatt wie im vorigen Abschnitt beschrieben Websites als Desktop-Programme zu nutzen, lässt sich ein Desktop auch als Website

darstellen.

Projekte wie *eyeOS* (<http://eyeos.org>) und *Online Operating System (OOS)*[OOS] zeigen, was damit gemeint ist. Es wird sich zeigen, ob dieser Ansatz erfolgreich ist. Es hört sich zwar interessant an, seinen individuell konfigurierten Desktop überall dabei zu haben, allerdings braucht man normalerweise keinen Desktop sondern bestimmte Programme.

Ein weiterer Schritt könnte Grid-Computing werden. Dies ist ein sehr weites Feld. Im Prinzip geht es hierbei darum, sich aus vielen verteilten Rechnern einen virtuellen Rechner bei Bedarf zusammenstellen zu können. Hardware wird flexibel mietbar.

Beispielsweise könnte eine Firma, die nur ab und zu Hardware Konstruktionszeichnungen anfertigen muss, sich für ein CAD-Projekt richtig viel Rechenleistung mieten, anstatt sich eine teure Grafik-Workstation zu kaufen. (Das passt auch zu den zweifelhaften Praktiken, Zeitarbeitsfirmen zu benutzen anstatt Personal einzustellen...)

5. Web-Standards

5.1 Der richtige Zeitpunkt

David Clark vom MIT hat eine Theorie aufgestellt, die es alleine aufgrund ihres kreativen Namens verdient, an dieser Stelle erwähnt zu werden, die *Apokalypse der zwei Elefanten*. In Tanenbaums Standardwerk zu Computernetzwerken wird sie folgendermaßen beschrieben[Tanenbaum03][S.64]:

Wird eine neue Technologie entdeckt, finden eine Menge Forschungsaktivitäten statt und es wird viel Wissen produziert. Nach einer Weile sind die Grundlagen bekannt und das wissenschaftliche Interesse nimmt ab. Nach dieser ersten Welle – dem „ersten Elefanten“ – fängt die Industrie an, sich für die neue Technik zu interessieren und investiert viel Geld in die Umsetzung – der „zweite Elefant“. Im Idealfall liegt zwischen diesen beiden Aktivitätsmaxima genügend Zeit für eine gute Standardisierung. Falls nicht, kollidieren die beiden Elefanten und es kommt zur Apokalypse. Das bedeutet hier, dass die Standards entweder noch nicht fertig sind und jedes Unternehmen eigene proprietäre Lösungen verfolgt oder das Standardisierungsgremium in die Enge getrieben wird und ein schlechter Standard die Folge ist weil das nötige Problemverständnis noch nicht vorhanden war.

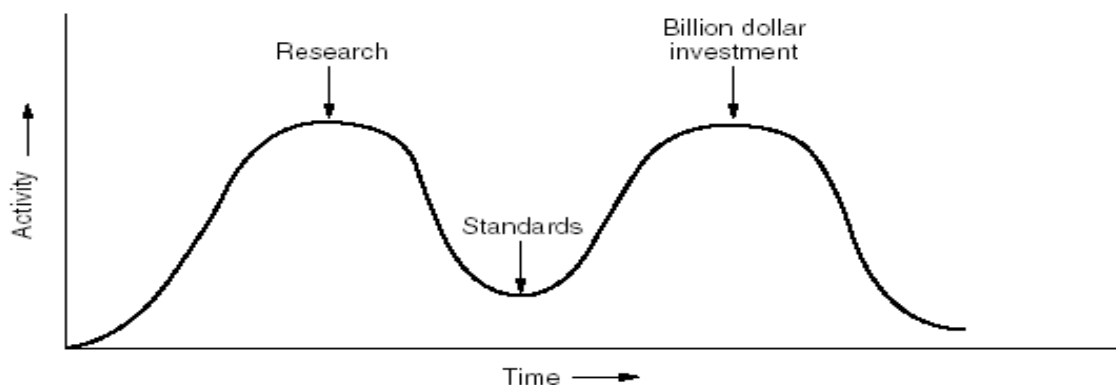


Figure 1-23. The apocalypse of the two elephants.

Abbildung 5.1: *Apokalypse der zwei Elefanten* (Scan aus der englischen Buchausgabe)

5.2 Das HTML der letzten beiden Dekaden

5.2.1 Inkompatible Browser

In der Anfangsphase des WWW handelten die Browserhersteller anscheinend nach der Devise „Standards sind toll – jeder sollte einen haben!“. Webseitengestalter müssen teilweise heute noch Inkompatibilitäten des Microsoft-Browsers berücksichtigen, da sie wegen Rückwärtskompatibilität einfach nicht dem Standard angepasst werden. Teil des Problems ist allerdings, dass das Browserverhalten nicht einfach korrigiert werden kann ohne dabei viele existierende Webseiten falsch darzustellen, da Webdesigner von der falschen Darstellungsmethode ausgehen.[Wilson08]

5.2.2 Plugins

Aktive Inhalte waren schon früh beliebt aber meistens problematisch. Mit aktiven Inhalten ist hier alles außer Text und Bildern gemeint. Bis heute werden sie oft als in Form proprietärer Plugins wie Java, ActiveX oder Adobe Flash realisiert.

Anwendungsgebiete können interaktive Physiksimulationen, Spiele, Abspieler für Ton und Filme und noch vieles mehr sein.

Plugins sind binäre Programme, die Daten im Browser darstellen, also nicht der Browser zeigt das YouTube-Video an, sondern ein Programm namens „Shockwave Flash Plugin“, dem man vertrauen muss. Dies ist mit einer Reihe an Nachteilen verbunden, z.B.:

- Flash-Filmchen sind nicht per Suchmaschine indizierbar (Allerdings arbeitet Adobe wohl an diesem Problem[Schonfeld08])
- Wenn der Hersteller einer runtime-Umgebung eine Plattform nicht unterstützen will, kann das Medium halt nicht benutzt werden (z.B. das Flash-Plugin auf 64-Bit-Linux)
- Potentielle Sicherheitslücken, auch in Hinblick auf Datensch(m)utz[FlashCookies]
- Barrierefreiheit – Flash-Inhalte sind auf Braille-Zeilen unbenutzbar
- Websites mit Videos nutzen mal bessere, mal schlechtere Abspiel-Applets, nicht alle laufen überall gleich gut (Siehe z.B. [Schmidt09]).
- Wenn das Plugin abstürzt, reißt es den Browser mit
- Das weit verbreitete Adobe Flash ist proprietär und kein offener Standard – Webseiten mit Flash-Applets sind streng genommen keine Webseiten, da sie nicht vom Webbrowser sondern von einem Programm im Browser interpretiert werden. Das ist technisch das selbe wie ein im Browser eingebetteter PDF-Betrachter.

Man könnte diese Problematik darauf zurückführen, dass das WWW sich Bedürfnissen gegenübergestellt sah, an die zur Zeit der ersten HTML-Standardisierungen niemand dachte. Damals gab es ein paar (meist langsame und instabile) Java-Applets, aber dass man Videos und Podcasts im Browser abspielen würde, war vor 15 Jahren nicht vorgesehen.

Um den höheren Bandbreiten und Rechnerkapazitäten gerecht zu werden, werden derzeit neue Standards und Softwaremodule entwickelt, welche glücklicherweise einheitlicher sind als ihre Vorgänger.

Auf der Software-Seite steht etwa mit webkit eine HTML-Renderengine zur Verfügung, die relativ flexibel und plattformübergreifend einsetzbar ist und deren Ergebnisse weithin für gut befunden werden.

Die für das „Web 2.0“ wichtigen Technologien (Auf Browserseite v.a. ein Subset von JavaScript und Flash-Videoplayern) konvergieren ebenfalls zu allgemeinen Lösungen. Da ist neben der Entwicklung freier Nachbauten (wie z.B. swfdec oder gnash) die direkte Einbindung von Multimedia-Inhalten in HTML5 zu nennen.

5.3 HTML 5

Zur Seite des Datenformats – HTML5 – lohnt es sich, ein paar mehr Worte zu verlieren, da dieser zukünftige Web-Standard die vorherigen Visionen unterstützt bzw. erst ermöglicht.

Nachdem am W3C⁶ häufig kritisiert wurde, dass die Weiterentwicklung seit HTML 4.01 und XHTML 1.0 zum Stillstand gekommen ist, haben Mitglieder einiger Unternehmen (Mozilla Foundation, Apple, Opera) 2004 eine Arbeitsgruppe mit Namen WHAT WG⁷ gegründet um diesen Zustand zu ändern[whatwg1].

HTML5 geht auf verschiedene Bedürfnisse heutiger Web-Autoren und Programmentwickler ein, von denen einige kurz beschrieben werden sollen.

5.3.1 Vorgeschriebenes Browserverhalten

HTML5 definiert, wie mit fehlerhaften Dokumenten umgegangen werden soll. Hier wurde aus der Vergangenheit gelernt, in der jeder Browser kaputten Code anders anzeigte und Webdesigner viel Aufwand (wie z.B. Browserweichen) treiben mussten, um ihre Seiten auf möglichst vielen Browsern möglichst ähnlich dargestellt zu bekommen. HTML5-konforme Browser sollten sich in Zukunft identisch verhalten.

Der HTML5-Standard bedient genau genommen zwei verschiedene Zielgruppen. Zum einen bekommen Web-Autoren vorgeschrieben wie neue, HTML5-konforme *Dokumente* auszusehen haben. Zum anderen ist festgelegt, welchen Input ein HTML5-konformer *Browser* wie darzustellen hat.

Das heißt, dass z.B. das <big>-Element nicht in HTML5-Dokumenten benutzt werden darf, Browser es aber dennoch auswerten müssen, um bestehende Webseiten weiterhin wie gewohnt anzuzeigen.

5.3.2 Verbesserte Semantische Dokumentauszeichnung

Neue Sprachmerkmale ermöglichen bessere Trennung von Form und Inhalt; so gibt es neue Tags, die aktuellen Nutzungsgewohnheiten Rechnung tragen. Durch das <nav>-Element kann ein Navigationsbereich vom Browser als solcher erkannt werden und z.B. immer schnell zugreifbar gehalten werden. Das ist z.B. für Mobilgeräte sinnvoll, welche per Tastendruck von Navigation auf Inhalt umschalten könnten.

Mit <article> könnten Weblog-Einträge, Foren-Postings, Zeitungsartikel und dergleichen ausgezeichnet werden.

Mit <time> können Datum und Uhrzeit sinnvoll repräsentiert werden. Dafür wird auch ein Datentyp definiert, so dass man nicht mehr raten muss, ob mit „03/05/10“ jetzt der 3. Mai 2010, der 5. März 2010 oder der 10. Mai 2003 gemeint ist.

Sinnvoll ist auch etwa <figure>, ein Element, das eine Beschriftung unter ein Bild oder Video setzt.

5.3.3 Einbettung von Multimedia

Mit <audio> und <video> sollte dem Wildwuchs an Codecs ein Ende bereitet werden, die oft lizenzrechtliche Probleme aufwerfen und auch technisch in der Regel sinnlos ist. Es sollte möglich sein, mittels diesen Tags direkt Mediendateien einzubinden, ohne sich von dubiosen Browser-Plugins wie Adobe Flash oder Microsoft Silverlight abhängig zu machen. Dazu wurde der freie Ogg Theora-Standard ausgewählt. Die Firma Apple torpedierte diese Idee allerdings um ihr eigenes QuickTime-Format nicht bedeutungslos werden zu lassen, und somit werden wir auch in der näheren Zukunft wohl mit vielen inkompatiblen Formaten zu kämpfen haben.[Meyer09]

Allerdings bringen diese beiden Multimedia-Tags trotzdem den Vorteil, dass die eingebetteten Dateien entweder mit der bevorzugten Abspielsoftware oder vom Browser selbst interpretiert werden können. Der vorläufige Standard sieht in Kapitel 3.14.9.10 Kontrollflächen vor zum Abspielen, Anhalten, Pausieren, zur

6 W3C: World Wide Web Consortium

7 WHAT WG: Web Hypertext Application Technology Working Group

Lautstärkenregulierung, zum direkten Anspringen einer Stelle, sowie einer Umschaltmöglichkeit zum Vollbildmodus[WHATWG07].

„Sollte sich so etwas, gepaart mit freien Medienformaten (Ogg Theora und Ogg Vorbis), durchsetzen, wären am Ende des Tages alle besser raus. Möglicherweise hat es sich dann bald ausgeflasht - zumindest in Sachen Video und Co.“[Schmidt09]

5.3.4 Eingebettetes SVG und MathML

Scalable Vector Graphics und *Mathematical Markup Language* sind zwei W3C-Standard-Auszeichnungssprachen für Vektorgrafiken und mathematische Formeln. Es ist vorgesehen, sie in HTML5 direkt zu unterstützen. So kann ein einfaches HTML5-Dokument mit einem grünen Kreis so aussehen[SVG]:

```
<!doctype html>
<title>SVG in text/html</title>
<p>
  A green circle:
  <svg> <circle r="50" cx="50" cy="50" fill="green"/> </svg>
</p>
```

5.3.5 Erweiterte Unterstützung von Web-Applikationen

HTML5 bietet wichtige Neuerungen für Webapps. So wurde die *webforms 2.0*-Spezifikation der WHATWG in HTML5 eingearbeitet. Webformulare mit Bedienelementen (Texteingabefelder, Buttons, Radiobuttons etc.) existieren schon seit jeher und sind auch notwendig um RIA-Anwendungen (Kapitel 4.3) zu ernsthaften Alternativen zu lokalen Programmen werden lassen. Der neue Standard erlaubt strenge Typprüfung und definiert einige Fälle, die bisher unklar waren. Dies erhöht die Stabilität und Plattformunabhängigkeit neuer Webapps.

Das `<canvas>`-Tag definiert eine Fläche, auf der per Javascript 2D-Grafiken erzeugt werden können. Diese sind auch beliebig nachträglich veränderbar. Damit ist von Diagrammen über Animationen bis hin zu komplexen Spielen alles möglich, was vorher nur unter Zuhilfenahme diverser Plugins realisierbar war. Auf <http://canvasedemos.com> können die Fähigkeiten bewundert werden.

Weitere APIs erlauben Drag & Drop, lokale Datenspeicherung für offline-Webapps, User-editierbare Inhalte, Cross-Document-Messaging und weiteres. [W3C09]

6. Mobile Geräte

Die Entwicklung von HTML5 hilft auch einer besonderen Art der Webnutzung, den Mobilgeräten. Nachdem sich die Idee einer abgespeckten Version des Internets in Form von WAP sich nicht richtig durchsetzen konnte, wird das nächste WWW darauf ausgerichtet sein, Form und Inhalt sauberer zu trennen.

Die in Kapitel 5.3.2 vorgestellte verbesserte Semantik wird endlich dazu führen, dass Einträge in Foren oder Blogs nicht mehr als blinde Tabellen mit fester Breite realisiert werden. Damit besteht die Hoffnung, sich in Zukunft Webseiten auf kleinen Bildschirmen ansehen zu können, ohne horizontal scrollen zu müssen.

Wohlgermerkt ohne dass sie auf großen Anzeigegeräten komisch wirken würden und ohne dass der Server unterschiedlichen Browsern angepassten Code senden müsste. Wobei letzteres durchaus gewollt sein kann. <http://mobil.bahn.de/> z.B. sendet erheblich weniger Daten als die normale Bahnseite um Übertragungskosten und Zeit zu sparen.

Aber auch ohne HTML5 wird das Netz bereits jetzt immer häufiger und selbstverständlicher mit mobilen Geräten benutzt. Diese Tendenz ist feststellbar an immer häufigeren Neuerscheinungen von Apple iPhone, Palm Pre, RIM BlackBerry, Nokia N810 und anderen Geräten, die ohne eine ständige Netzanbindung fast nutzlos sind.

7. Das semantische Web

Dieses Kapitel beschreibt die Vision „Das Web, das mich nicht mehr nervt“.

Die Hauptaufgabe des WWW, bzw. des gesamten Internets ist es, Informationen zu übermitteln. Die Methode wird immer effizienter. Zur Zeit der „Katalog-Websites“ wurden teilweise Texte als Bilddateien eingesetzt, damit die Seiten auch das corporate design mit gewünschten Schriftarten etc. enthielten und Werbeagenturen die Website, „Multimedia-CDROMs“ und Papier-Kataloge „wie aus einem Guss“ verkaufen konnten.

Das ganze war für Suchmaschinen natürlich nicht gut verwertbar und diese Seiten wurden schlecht gefunden. Andere Unarten dieser Kategorie beinhalten die Nutzung von Frames und Macromedia Flash.

Neben der geringeren Auffindbarkeit sperren solche „optimierten“ Seiten viele Nutzer aus, seien es PDA-Nutzer mit kleinen Bildschirmen oder blinde Nutzer mit mit Braille-Zeile anstatt Bildschirm.

Aber auch Informationen in Textform können recht schwierig zu finden sein, weil die Begriffe entweder nicht eindeutig beschreibbar sind oder einfach zu viel zu einem Thema existiert und z.B. die gesuchte Nachrichtenmeldung unter tausenden Treffern in der Suchmaschine einfach untergeht.

Auch sind Bilder, Videos und Audio-Dateien per konventioneller Textsuche unauffindbar, sofern keine Abschrift oder beschreibender Text vorhanden ist.

Kurz gefasst: Es ist oft schwierig, gesuchte Informationen schnell zu finden.

Und die Lösung ist der Kern des *semantischen Webs*: Informationen sollten so gut wie möglich automatisch interpretiert werden können, also *maschinenlesbar* vorliegen.

7.1 Wie Semantik unser Leben erleichtern könnte

Die Vorteile sollen anhand eines Beispiels beschrieben werden, dessen Technologien bereits vorhanden sind, allerdings noch nicht konsequent und flächendeckend eingesetzt werden.

HTML wird mit XML-Erweiterungen für alle möglichen Themengebiete Kategorisierungen von Informationen erlauben. Z.B. können mit der Erweiterung ThML (Theological Markup Language) Bibelreferenzen (Z.B. Johannesevangelium, Kapitel 3 Vers 16) beschrieben werden.

Andere Auszeichnungen identifizieren Orte und Zeiten eindeutig.

Die Vision des semantischen Webs nur mit diesen drei Erweiterungen erlaubt folgendes Szenario:

Ich bin für ein Wochenende in einer fremden Stadt und möchte am Sonntag in die Kirche gehen. Ich öffne die Webseite einer Kirchengemeinde und dort den Veranstaltungskalender. Am entsprechenden Sonntag existiert eine Schaltfläche „eintragen“, auf die ich klicke.

Daraufhin wird in meinem Kalender ein neuer Termin am richtigen Zeitpunkt eingetragen. Diesem Eintrag hängt eine Anfahrtsskizze zur Kirche an, welche automatisch aus den Ortskoordinaten erzeugt wird. Diese Skizze hat aber nicht der Webmaster der Kirchenhomepage vorher abgelegt, sondern das Eintragungsprogramm würde sie mit einem Kartendienst (openstreetmap, Google maps etc.) mithilfe der eingebetteten Koordinaten selbst verknüpfen. Voreingestellte Parameter (Ich bin Fußgänger, Radfahrer oder Autofahrer, regnet es? Wenn ja, welchen Bus kann ich nehmen?) könnten dabei berücksichtigt werden.

Abgerundet wäre der Kalendereintrag mit Zusatzinformationen, wie im Beispiel einer Referenz auf die Bibelstelle von welcher voraussichtlich die Predigt handeln wird; ein Klick darauf würde mein Bibelprogramm (Lokal installiert oder Webapp) starten, in dem der entsprechende Predigttext markiert wäre. Weitere Informationen könnten E-Mail-Adresse des Pfarrers sein, ob die Kirche Rollstuhl-geeignet ist, ob Kinderbetreuung angeboten wird etc.

Das wäre jetzt *Semantik*. Der Computer kann anhand geeigneter *Metainformationen*⁸ Daten sinnvoll für mich

⁸ altgr. Μετά: mit, bei. Metadaten sind also Daten, die Daten beschreiben. Etwa, dass der String „04.07.“ ein Kalenderdatum ist.

verarbeiten. Vernetzt – als *semantisches Netz* – werden diese maschinenlesbaren Informationen noch nützlicher:

Hätten alle Kirchen ihre Websites mit semantischen Auszeichnungen angereichert, könnte eine Suchmaschine diese sammeln und man erhielte auf Wunsch automatisch generierte Listen, die z.B. alle englischsprachigen Gottesdiensten enthielten, die am nächsten Sonntag zwischen 11:00 und 13:00 Uhr im Radius von 20km um Frankfurt herum stattfinden und sich um die Bergpredigt drehen.

Ich müsste also nicht bereits die Kirchenwebseite aus dem Beispiel kennen, sondern würde mein persönliches Suchprogramm (als Web-Applikation auf meinem Server oder mit meinem persönlichen Profil bei einem öffentlichen Dienst – sog. *Web-Agent*) öffnen und es einfach fragen.

Diese Informationen über Textstring-basierte Suchmaschinen zusammenzutragen ist derzeit nicht möglich, bzw. nicht automatisch.

Andere Suchanfragen könnten sein: „Wo finde ich in Hagen Bäcker mit glutenfreien Brötchen und bis wann haben die heute geöffnet?“ oder „Wer will mit mir Go spielen?“. Die Antworten laufen wohlgerne nicht über ein Portal für Bäcker, das händisch gepflegt wird oder eine Kontaktbörse mit Go-Spielern, sondern wird automatisch generiert.

Mit anderen Worten: Computer verstehen die Informationen und generieren daraus neues Wissen.

7.2 Verschlagwortung von Daten

Eine der größten Herausforderungen besteht darin, komplexere Inhalte maschinell verwertbar zu machen. Während Informationen wie Ortskoordinaten und Termine noch recht einfach einzuordnen sind, sieht es mit Zeitungsartikeln, Bildern, Podcasts und Videos anders aus. Computer „sehen“ andere Dinge als Menschen. Darauf wird im Kapitel 7.4 noch einmal im Bezug auf Webseiten eingegangen.

Wenn jetzt beispielsweise ein Zeitungsredakteur einen Reisebericht mit einem passenden Bild illustrieren will, muss er in der Lage sein, nach gewünschten Inhalten zu suchen. In einem nicht-semantischen Web ist es ein Problem, Bilder zu suchen, die beispielsweise einen schlafenden Hund in einem Liegestuhl am Strand zeigen. Dieses Problem existiert allerdings für Fotoagenturen schon seit Jahrzehnten und wurde bereits gelöst. Die Lösung heißt *Verschlagwortung*.

Dieses geschieht bei Fotografien durch Metadatenfelder, die im Dateiheder eingebettet werden. Neben Schlagworten können auch noch Daten über den Autor, eine Inhaltsbeschreibung, ein Titel, Informationen über Entstehungszeit und -ort und anderes erfasst werden (Dublin Core-Standard, siehe unten).

In den 70ern wurde dazu von der IPTC⁹ und NAA¹⁰ ein Standard geschaffen, der auch heute noch gebräuchlich ist, heute vor allem in „IPTC-Feldern“, die man in seine Bilddateien einfügen kann [IPTC]. Dieses Format heißt IIM¹¹ und kann prinzipiell alle Arten von Daten (Bild, Ton, Video, Text) beschreiben.

IPTC-Felder wurden 1994 eingeführt, zu einer Zeit bevor XML¹² seinen großen Aufstieg erlebte. XML ist ein Format, welches viele Vorteile bietet. Es ist frei von Patenten, einfach zu handhaben, plattformunabhängig und erweiterbar. Es wird auch vom W3C für Datenaustausch und Metadaten empfohlen (RDF, siehe weiter unten). 2001 stellte Adobe als Nachfolger XMP¹³ vor, eine Mischung aus XML und RDF [Riecks].

Im Zuge der letzten Jahre wurde Verschlagwortung – englisch: *tagging* – im „Web two null“ verstärkt für alles mögliche und unmögliche eingesetzt. Damit ist möglich, alle Blogeinträge, YouTube-Videos und flickr-Bilder zu finden, die mit den Begriffen *Hund*, *schlafen*, *Sonnenstuhl*, *Strand* assoziiert sind.

Probleme sind verschiedene Sprachen, Konventionen (sollte man immer Mehrzahl benutzen?), Schreibfehler, Interpretationen und persönliche Ansichten (Welche Tags passen auf bestimmte Politiker?).

9 IPTC: International Press Telecommunications Council

10 NAA: Newspaper Association of America

11 IIM: Information Interchange Model

12 XML: eXtended Markup Language

13 XMP: eXtensible Metadata Platform

Eine Lösung stellt eine endliche Menge von Wörtern dar, ein *controlled vocabulary*[Riecks2]. Hiermit wird sichergestellt, dass ein Konzept mit genau einem Begriff beschrieben wird. Problematisch dabei ist, dass sich jeder an ein vorgegebenes Vokabular halten muss. Dies funktioniert in Bibliotheken mit geschultem Personal, nicht aber im freien Internet. Allerdings könnte z.B. in einer Medizinischen Datenbank die Eingabe „Herzinfarkt“ nach Rückfrage automatisch umgewandelt werden in den kanonischen Begriff „Myokardinfarkt“. Es geht hier ja schließlich um Technologien der Zukunft...

Eine wichtige Sammlung von Metadaten-Elementen soll hier aber noch erwähnt werden. 1995 wurde in Dublin/Ohio das *Dublin Core metadata set* definiert und ist inzwischen ein ISO-Standard (15836). Hier sind 15 Elemente festgelegt zur Beschreibung von Dokumenten. Dokumente sind hierbei Webseiten, Bilder, Videos, Ton, und alles andere. Benutzt wird das Dublin Core set unter anderem in dem oben beschriebenen XMP, RSS-Feeds und OpenDocument-Dateien. Die 15 Elemente heißen: Contributor, Coverage, Creator, Date, Description, Format, Identifier, Language, Publisher, Relation, Rights, Source, Subject, Title und Type. Die Bedeutung ist eindeutig beschrieben. Jedes Element darf mehrmals vorkommen aber auch weggelassen werden.[DC08]

Ein Problem besteht mit Schlagworten jedoch noch: Sie sind nicht eindeutig. Die englische Literatur spricht von *concept*, einem Wort das man „wordly overseaten“ könnte als *Konzept*, korrekt aber wahrscheinlich eher als *Begriff* übersetzt wird.

Mit Begriff ist etwas gemeint, das ein Bild in unserem Kopf auslöst. Also *schnell* bedeutet z.B. „hohe Geschwindigkeit“. Auch gibt es mehrere Städte mit gleichem Namen. Bei *Paris* denken die einen an eine Stadt in Texas, andere vermuten sie in Arkansas, Idaho, Californien, Illinois, Kentucky, Maine [...] oder sogar Frankreich[WP09].

Ein *Begriff* ist dagegen eindeutig, da er nicht das Wort sondern die Idee darstellt.

Für Orte kann dies z.B. mit dem Dienst *geonames* bewerkstelligt werden hier sind derzeit über 6,5 Millionen Ortsangaben mit eindeutigen URLs verknüpft[geonames]. Hagen kann etwa eindeutig beschrieben werden mit der URL <http://sws.geonames.org/2912621/>.

7.3 Verknüpfung von Daten

Begriffe können katalogisiert, sortiert und gefunden werden. Dies ist der erste wichtige Schritt für das semantische Web. Der zweite bringt das Wissen miteinander in Beziehung.

Ein paar Beispiele für Beziehungen:

- Dreiräder sind Fahrzeuge, aber nicht alle Fahrzeuge Dreiräder.
- „Geschäft“ ist ungefähr das gleiche wie „Laden“, „Supermarkt“, „Shop“, „Kiosk“, „Kaufhaus“ und „Warenhaus“
- Die Schwippschwägerin meiner Mutter ist die Tochter des Vaters meines Vaters und damit meine Tante.
- Akihabara ist ein Stadtteil Tokios in der Ward Chiyoda, welches wiederum Teil Tokios ist, der Hauptstadt von Japan.
- Tokio und Washington sind beides Hauptstädte.
- Die Mona Lisa ist ein Bild, das von Leonardo da Vinci gemalt wurde

Solche Beziehungen lassen sich in Form eines Netzes aus Graphen abbilden, das *Ontologie* genannt wird. Die Wissensrepräsentation ist Teilgebiet der Forschung über Künstliche Intelligenz und wurde mit der Diskussion des semantischen Webs wieder recht aktuell. Eine ausführliche Beschreibung gibt es auf den Webseiten der Gesellschaft für Informatik e.V.[Hesse02].

Ein Wissensnetz zielt darauf ab, aus Verknüpfungen zwischen den einzelnen Begriffen automatische Schlussfolgerungen zu ermöglichen. Die Aussage „Die Mona Lisa wurde gemalt von Leonardo da Vinci“ impliziert z.B. „Leonardo da Vinci ist ein Maler“ und „Leonardo da Vinci hat die Mona Lisa gemalt“.

7.4 Technische Umsetzung des semantischen Webs

Das WWW ist in erster Linie für Menschen gemacht. Die benutzten HTML-Auszeichnungselemente beschreiben die Struktur eines Dokuments, sagen aber nichts über den Inhalt aus. Ein Mensch erkennt die Hauptüberschrift als Artikel-Titel und die Überschrift zweiter Ordnung als Namen des Autors. Der Computer erkennt dies nicht, sondern lediglich den Dokumentaufbau. Der Unterschied ist groß, wie Abbildung 4.1 verdeutlicht.



Abbildung 7.1: Links: Was der Computer sieht, rechts: Was der Mensch sieht.[W3C08]

Gewünscht wird, dass der Computer die gleichen Informationen versteht wie der Mensch. Es gibt zwei prinzipielle Methoden, dies zu erreichen. Die „leichtgewichtige“ Variante in Form sogenannter *Mikroformate* versucht dabei, Redundanz zu vermeiden, die „schwere“ hat zum Ziel, ein Themengebiet umfassend mit Ontologie-Beschreibungssprachen zu erfassen. Beide Varianten sollen vorgestellt werden.

7.4.1 Mikroformate

Die Grundidee der Mikroformate ist, das existierende Web nicht abzulösen, sondern bestehende Elemente mit beschreibenden Hinweisen anzureichern. Also das `<h3>`-Tag wird beispielsweise mit einer Anmerkung ausgestattet, die besagt, dass es den Autor-Namen umschließt.

Diese auch „lower case semantic web“¹⁴ genannte Techniken versuchen also nicht die gesamte Welt geschlossen zu beschreiben, sondern das existierende Web um kleine semantische Elemente anzureichern[çelik04]. Dazu werden selten genutzte HTML-Attribute verwendet („überladen“), die bereits existieren, normalerweise *class*, manchmal auch *rel* und *rev*.

Im folgenden Beispiel beschreibt *fn* den vollen Namen und *org* die Organisation, der die Person angehört.

```
<span class="vcard">
  <span class="fn">Jeremy Keith</span>,
  <span class="org">Clearleft</span>
</span>[MFwiki]
```

¹⁴ Also „semantic web“ anstatt „The Semantic Web“, das alles umkrepeln soll

7.4.2 RDFa

Einen ähnlichen aber etwas anderen Ansatz beschreibt das W3C. Es empfiehlt für diesen Zweck eine Sprache namens RDFa¹⁵, mit der XHTML um semantische Attribute *erweitert* werden kann. XHTML und nicht HTML, da ersteres erweiterbar ist. HTML5 macht hier selbstverständlich keine Probleme mehr.

Im obigen Beispiel (Abbildung 4.1) sieht das wie folgt aus:

Während „klassisches“ (X)HTML lediglich die Struktur beschreibt...

```
<div>
  <h2>The trouble with Bob</h2>
  <h3>Alice</h3>
  ...
</div>[W3C08]
```

...ist mit den RDFa-angereicherten Tags klar, worum es geht:

```
<div xmlns:dc="http://purl.org/dc/elements/1.1/">
  <h2 property="dc:title">The trouble with Bob</h2>
  <h3 property="dc:creator">Alice</h3>
  ...
</div>[W3C08]
```

Das „dc:“ wählt einen *namespace* aus, besagt also dass in diesem Fall das Dublin Core-Vokabular zugrunde liegt. Namensräume sind sinnvoll, um verschiedenen Kontexten gerecht werden zu können. So könnte „title“ in Adressverzeichnissen etwas anderes beschreiben als der hier gemeinte Titel eines Artikels. Das Namensraumkonzept erlaubt es jedem Veröffentlicher, eigene Auszeichnungen zu erfinden (Wie sinnvoll das wiederum wäre, ist eine andere Frage).

Was ist jetzt der Unterschied zwischen Mikroformaten und RDFa?

Microformats ist eine Initiative, die von Technorati gestartet wurde und schnell von vielen Bloggern übernommen wurde. Sie ist derzeit ein de-facto-Standard, der zwar funktioniert, aber nicht so gut durchdacht ist und sich daher auch als weniger skalierbar erweisen wird als RDFa.

Ein großes Problem ist, dass Mikroformate sowohl die Syntax als auch ein paar Vokabeln definieren für häufige Anwendungen wie Datum, Ortskoordinaten, Lizenzen etc. Mit anderen Worten: Mikroformate kennen keine Namensräume und Zweideutigkeiten werden früher oder später zwangsläufig auftreten. Das Format ist nicht gut erweiterbar.

RDFa dagegen wird seit 2004 diskutiert um am Ende einen guten verbindlichen Standard zu haben. Seit 2008 hat er den Status einer *Empfehlung* und es existiert ein *Primer* ([W3C08]). RDFa kümmert sich um eine saubere Syntax und lässt das Vokabular offen. Dadurch kann *jedes* Vokabular benutzt werden[Birbeck08]. Im letzten Beispiel etwa fand Dublin Core¹⁶ Anwendung.

Im Prinzip haben wir hier wieder ein Elefantenapokalypsenproblem¹⁷; Microformats funktioniert schon heute irgendwie und ist bereits verbreitet. RDFa ist zwar besser, kam aber später. Dazu kommt, dass Mikroformate in HTML4 funktionieren, während RDFa XHTML oder HTML5 benötigt.

Die BBC hat aufgrund mehrerer Schwächen der Microformats 2008 angekündigt, diese auf ihren Webseiten durch RDFa zu ersetzen[BBC]. Wahrscheinlich wird RDFa sich als Standard durchsetzen.

Weiterführende Informationen zu der Spaltung des „lower case“-semantic Webs in die beiden Formate beschreibt[Prodromou08].

15 RDFa: Resource Description Framework in attributes

16 Kapitel 7.2 unten

17 Kapitel 5.1

7.4.3 Upper case semantics

Die „lower case semantics“ haben den Vorteil, dass sie recht schnell sichtbare Ergebnisse bringen, aber sie schöpfen das volle Potential des Semantischen Webs bei weitem nicht aus. Will man die Beziehungen aus Kapitel 7.3 ausdrücken, dann muss eine Ontologie, also ein Wissensnetz aufgebaut werden.

Das von W3C empfohlene Format RDF¹⁸ kennt für diesen Zweck sogenannte *triplets*, zu deutsch *Tripel*.

Ein Tripel ist dabei ein einfacher Ausdruck, der aus <Subjekt> – <Prädikat> – <Objekt> besteht.

Zum Beispiel: <Leonardo> <malte> <Mona Lisa>.

Wenn jedes Tripel-Element ein eindeutiger Begriff ist, lassen sich damit schon viele Relationen darstellen.

RDF definiert dabei allerdings nur, dass Aussagen als Tripels formuliert werden sollen. Wie sie *serialisiert*, also tatsächlich als Dateien auf die Festplatte gelegt werden sollen, ist freigestellt. Neben dem bereits vorgestellten XML (z.B. in Form von RDFa als Erweiterung von XHTML) können auch N3 oder Turtle zum Einsatz kommen.[W3C98]

Um Ontologien aufzubauen, entwickelt das W3C eine eigene Sprache namens OWL¹⁹, inzwischen OWL2[OWL2]. Diese benutzt die Syntax von RDF.

Hier wird allerdings noch intensiv geforscht und viele Sachen sind sowohl sehr kompliziert als auch noch veränderlich.

Für RDF existieren allerdings bereits funktionierende Anwendungen, etwa *Creative Commons*, welche Lizenzen beschreiben oder FOAF²⁰, einer Art verteiltem *social network* (<http://www.foaf-project.org/>).

8. P2P-Technologie

8.1 Offene Entwicklung

Fertige Hardware wird einerseits immer billiger, andererseits bieten hochintegrierte Einzelkomponenten (SoC – System on a Chip) auch Hobbyisten die Möglichkeit, leistungsfähige Computer selbst zu bauen.

In den 60ern und 70ern war Software innerhalb der Unix-Hackergemeinde prinzipiell frei, man verteilte sie mit Magnetbändern an befreundete Uni-Arbeitsgruppen und jeder konnte sie benutzen. Mit der DOS-Ära gingen diese Traditionen eher verloren, man war mit „freeware“, also kostenlosen ausführbaren Programmen ohne Quellcode zufrieden. Mit der Open Source Software-Bewegung ist dieser Missstand inzwischen wohl wieder berichtigt worden.

Meine Prognose: Mit weiterer Entwicklung und einer „kritischen Masse“ an Community wird Hardware immer offener und teilweise auch selbst entwickelt werden.

Beispiele: MP3-Player-Firmware „Rockbox“: Ursprünglich ein reines Fanprojekt ohne Unterstützung von Archos (Dem Player-Hersteller), inzwischen das wohl beste Betriebssystem für die unterstützten Plattformen, die es gibt. (<http://www.rockbox.org/>)

Openmoko ist eine kommerzielle Hardware für Mobiltelefone, die mit offenem Linux-Betriebssystem betrieben wird. (<http://openmoko.org>)

Die tragbare Spielekonsole OpenPandora stammt aus der Community, Hard- und Software komplett frei (wobei Hardwarepläne allerdings aus Angst vor Plagiaten unter Verschluss gehalten werden...) (<http://openpandora.org/>)

Andere interessante Projekte wie z.B. die Software kismet zum „Wardriven“ konnte zur Zeit der ersten Verbreitungswelle von 802.11 auch nur aus der Nutzerschaft kommen und nicht von Marketingstrategen.

18 Resource Description Framework

19 OWL: Web Ontology Language, der Buchstabendreher (man würde WOL erwarten) ist beabsichtigt.

20 FOAF: Friend Of A Friend

8.2 Drahtlose Mesh-Netzwerke

Auf diese Weise werden sich möglicherweise Netzwerke aus billiger, offener (und damit unbeschränkbarer) Hardware bilden, welche per Funk miteinander kommuniziert, ohne auf Infrastruktur angewiesen zu sein.

Technisch basiert diese *Mesh-Technik* bedeutet, dass Hosts selbstständig Pakete weiterleiten, wenn sie zwischen anderen Hosts sind, welche sich einander nicht direkt erreichen können. Das Internet ist ein gutes Beispiel dafür.

Drahtlose Mesh-Netzwerke basieren meist auf dem *ad-hoc-Modus* von 802.11-Geräten (WLAN). Dies ist für viele Regionen auf der Welt die einzige Möglichkeit, überhaupt Internetanschluss zu bekommen (Entwicklungsländer) oder zumindest zu akzeptablen Bedingungen (Preis, Bandbreite).

Die benötigten Routing-Protokolle sind noch immer in der Entwicklung, derzeit wird häufig noch OLSR²¹ eingesetzt, die Weiterentwicklung heißt B.A.T.M.A.N.²²

Das wahrscheinlich bedeutendste Beispiel ist die Freifunk-Initiative. (<http://freifunk.net>)

Ein sehr empfehlenswertes Buch zu dem Thema ist „Wireless Networking in the Developing World“ und kann frei von der Projekthomepage heruntergeladen werden. (wndw.net)

Anwendung findet die Idee in dem (davon unabhängigen) One-Laptop-per-Child-Projekt (<http://laptop.org>).

8.3 Mögliche Realisierungen offener Hardware

Abgesehen von offener Firmware und Betriebssystemen, die auf erschwinglichen aber industriell hergestellten Computern läuft, werden Privatleute sich in mittelfristiger Zukunft auch Hardware selbst herstellen oder zumindest auf Ihre Bedürfnisse hin modifizieren können.

Dazu fallen mir drei aktuelle Projekte bzw. Produkte ein.

1. Das RepRap-Projekt

Hierbei handelt es sich um ein Hobby-Projekt. RepRap ist ein 3D-Drucker, der sich selbst replizieren kann. Der Name RepRap kommt von **Re**plicating **Ra**pid-prototyper. (<http://reprap.org>)

2. Das Beagle-Board (<http://beagleboard.org/>) (Stellvertretend für mehrere leistungsfähige DSP-Boards)

Das Beagle-Board ist zwar in gewisser Hinsicht kommerziell hergestellt und vertrieben, allerdings ist es komplett frei, Bastler und Ingenieure in den Entwicklungsabteilungen diverser Firmen können sich für 149 US\$ ein kleines System kaufen mit Laptop-ähnlichen Leistungsdaten.

Mit entsprechender Expertise ist es möglich, sich daraus einen PDA herzustellen, der viele mobilen Bedürfnisse abdeckt.

3. nVidia CUDA (Compute Unified Device Architecture)[nvidia]

Für 3D-Simulationen und -Spiele gibt es seit langem 3D-Grafikbeschleuniger. Entwickler wünschen sich Physik-Beschleuniger-Hardware, die jedoch nicht genug verkauft werden könnten.

Der Grafikkartenhersteller nVidia hat eine Parallel-Processing-Plattform entwickelt mit frei konfigurierbarer Hardware (Entfernt vergleichbares Prinzip wie FPGAs) und einer standardisierten API. Diese Plattform wird bereits in den aktuellen High-End-Grafikkarten verbaut.

Diese Hardware erlaubt für bestimmte Algorithmen die Rechenleistung eines Großrechners auf einer PC-Einsteckkarte.

Neben Wetter- und Atombombensimulationen fällt ein Anwendungsbereich ein, welcher direkt das Web von morgen betrifft: Cryptographie und Cryptoanalyse.

Meine Prognose: Das wird noch ein großes Thema! (Privatsphäre, e-Banking, Spionage etc.)

²¹ OLSR: Optimized Link State Routing

²² B.A.T.M.A.N.: Better Approach To Mobile Adhoc Networking

9. Risiken

Das Internet und das WWW haben in den letzten zehn Jahren die Gesellschaft wohl mehr verändert als kaum eine andere Technologie vorher. Natürlich, das Rad war auch eine recht wichtige Entdeckung, aber die Menschen konnten sich Jahrzehnte lang an dieses neue Konzept gewöhnen und es ist auch relativ leicht zu erfassen. Die Auswirkungen auf das menschliche Leben kamen eher gemächlich daher. Das Internet hingegen birgt Möglichkeiten und Risiken, die zum großen Teil wahrscheinlich auch heute noch unbekannt sind und stellt viele althergebrachte Gewohnheiten auf den Kopf.

9.1 Überwachbarkeit

Vor dem gläsernen Menschen wird schon lange gewarnt. Die heutige Gesellschaft hat noch nicht gelernt, mit dem Netz umzugehen, welches nicht vergisst. Freigiebig werden soziale Netze mit persönlichen Daten gefüllt und Zeitungen sind voll mit Artikeln über Personalchefs, die Bewerber mit peinlichen Fotos konfrontieren, die z.B. Parties und Alkohol zum Inhalt haben.

Auf staatlicher Seite wird ebenfalls eine Datensammelwut und ein Kontrollwahn ausgelebt, wie es für demokratische Gesellschaften unglaublich ist. Sei es Vorratsdatenspeicherung, Videoüberwachung des öffentlichen Raumes, Einführung einer Infrastruktur zur Internetzensur („Zensursula“) oder der „Bundestrojaner“ – die Freiheit wurde in den letzten Jahren stetig mehr beschnitten.

Datenschutz und informelle Selbstbestimmung muss aktiv erkämpft werden. Der Trend wird sich vermutlich fortsetzen.

Starke Kryptographie und Anonymisierungstechniken (TOR, Darknets[Welchering09] etc.) werden entweder notwendig für das tägliche Leben oder verboten. Oder beides.

9.2 Informationsüberflutung

Dieses Phänomen ist nicht neu, wird aber von immer mehr Menschen quälend wahrgenommen: Die Informationsberge wachsen täglich, das eigene Gehirn aber nicht.

E-Mail-Postfächer quellen über und wachsende Zeitschriftenstapel erzeugen bei vielen Menschen Gefühle der Ohnmacht.

Eine aktuelle Methode, Informationen des WWW zu kanalisieren ist der RSS-Reader.

Ein persönliches Beispiel dazu: In meinem RSS-Reader befinden sich derzeit 193 Feeds mit insgesamt 14518 ungelesenen Artikeln. Technik alleine hilft also nicht. Man könnte zwar einwenden „Das beste Werkzeug ist nur Tand in des tumben Toren Hand“, aber ich denke das greift zu kurz. Das Grundproblem ist dass die Informationsmenge für eine (interessierte) Person einfach zu groß ist.

Und die muss verarbeitet werden. In den Worten Wau Hollands ist dies der Prozess, aus Daten, die in „Bit“ gemessen werden, Wissen zu produzieren, welches mit „Deut“ quantifiziert werden kann. Er betonte auch gerne, dass das Wichtigste Element eines Input/Output-Systems der Schrägstrich ist.

Die Kunst besteht also darin, in der Flut aller neuen Informationen die relevanten heraus zu filtern. Es werden Filtersysteme nötig. Eine Redaktion ist beispielsweise solch ein Filter. Oder auch Plattformen wie technorati, die neue Blog-Einträge sammeln und nach Beliebtheit sortieren.

Das semantische Web könnte eine weitere Hilfe werden, indem unbenötigte Informationen gezielter aussortiert werden ohne zu viele *false negatives* zu erzeugen.

Allerdings ist dies wahrscheinlich eher ein gesellschaftliches Problem, welches mit technischen Mitteln nur unzureichend bekämpft werden kann; der Mensch muss lernen, dass er nicht alles Wissen muss und kann, was ihm angeboten wird.

Vielleicht ist dies vergleichbar mit dem Nahrungsangebot in westlichen Industrieländern. Während in früheren Zeiten Hungersnöte mehr oder weniger regelmäßig ausbrachen, war es weise, sich einen großen

Vorrat anzulegen und alles zu essen was man bekam. Heute existieren Lebensmittel im Überfluss und würde man alles essen, was man essen könnte, wäre das sehr ungesund...

Wichtig ist die richtige Auswahl. Der Körper benötigt regelmäßig gesunde Nahrung in der richtigen Menge, um ordnungsgemäß zu funktionieren, also muss aus dem überreichen Angebot die entsprechende Teilmenge ausgewählt werden.

Mit der gesellschaftlichen Weiterentwicklung wird sich möglicherweise die Erkenntnis durchsetzen, dass es dem Gehirn ebenfalls besser geht, wenn es mit wenigen aber wertvollen Informationen gefüllt wird. Sichtbar wird dieser Fortschritt an dem Tag, an dem alle Fernsehstudios und Redaktionen von Boulevardzeitungen geschlossen werden und die Leute wieder anfangen richtige Literatur zu lesen...

Meine Zukunftsprognose: Das dauert noch sehr, sehr lange.

9.3 Spam

Eine besondere Art von zu viel Information stellt Spam dar. Spam, also unerwünschte Flutung eines Kommunikationsmediums mit Werbung oder anderen sinnfreien Inhalten bedroht das Web. Auch andere Medien leiden darunter, aber während man sich über Anrufe und berstende Briefkästen mal ärgert und evtl. mit Aufklebern und Beschwerden bei der Bundesnetzagentur sogar etwas dagegen tun kann, wird ein zugespantes Wiki oder Forum schnell unbenutzbar.

Ein Beispiel dafür ist die Trackback-Funktion in Weblogs, die eine automatische und unkomplizierte Vernetzung der „Blogosphäre“ erlauben soll. Nachdem Firmen dies für Suchmaschinensпам missbraucht haben, wurde das nofollow-Meta-Attribut eingeführt, das Suchmaschinen anweist, Links nicht in ihrem Index zu berücksichtigen. Dieses Element zerstört also zur Spam-Abwehr direkt das Netz.

Das semantische Web wäre je nach Topologie natürlich ebenfalls stark anfällig dafür. Selbst wenn das Konzept „Spam“ ontologisch beschrieben wäre, bliebe dies gefährlich, da Angreifer missliebige Inhalte damit einfach als solchen diskreditieren könnten. Da diese Problematik so allgegenwärtig und brennend wichtig ist, werden dafür wahrscheinlich leistungsfähige Lösungen entwickelt werden.

Im übrigen hat selbst spam auch seine guten Seiten, wo kann man innerhalb eines Jahres sonst das alles gewinnen?

”

- 24.557.550€ an Geldgewinnen
- 13.220€ an verspielbaren Gewinnen
- 7 neue PKW
- 87 Testpackungen Viagra und ähnlichem
- 189 Dates mit willigen Frauen
- 15 Dates mit willigen Männern
- 2 Kreuzfahrten
- 1 Seeligsprechung [sic]“[Zappi07]

10. Fazit

Wie wird das Web der Zukunft aussehen? Es liegt in der Natur der Sache, darauf keine definitive Antwort geben zu können.

Sicher erscheint mir aber, dass es eine wesentlich wichtigere Rolle spielen wird als heute. Netzbasierte Programme werden viele Aufgaben heutiger Desktop-gebundener Software übernehmen. Software wird nicht mehr an einen Computer gebunden sein, sondern von überall. Insbesondere wird sie von mobilen Geräten *gut* nutzbar sein.

Die Ideen des semantischen Webs werden vermutlich noch eine bis mehrere Revolutionen auslösen.

Viele experimentelle Techniken werden standardisiert, wenn sie für gut befunden werden. Das ist eine der schönen Seiten des Internets: Es wird von einer lebendigen Nutzerschaft getragen, was die Standardisierungsgremien einschließt. Dies erkennt man daran, dass sie sich nicht ohne Grund gegen etablierte Trends stellen²³. Dieses Verhalten tut dem Fortschritt und der Interoperabilität gleichzeitig gut.

Ob die menschlichen und sozialen Probleme wie Datenschutz, Informationsüberlastung und willkürlicher Missbrauch technisch gelöst werden können ist fraglich. Wie lange die Gesellschaft braucht, um sich auf das Web der Zukunft einzustellen, ist vermutlich einer der interessantesten Aspekte des WWW der Zukunft.

Es bleibt spannend.

²³ So bleibt etwa das häufig eingesetzte `<small>` in HTML5 erhalten, während das (semantisch) genauso sinnfreie `<big>` herausfliegt aufgrund zu seltener Nutzung.

Literaturverzeichnis

- Simmons07: Simmons, Dan, , 2007, http://www.dansimmons.com/news/message/2007_01.htm
- WP01: Wikipedia-Autoren, siehe Versionsgeschichte, Seite „Optische Telegrafie“ , , http://de.wikipedia.org/w/index.php?title=Optische_Telegrafie&oldid=63048502
- /09: Oracle Goddess, Amazon Pulls Purchased E-Book Copies of 1984 and Animal Farm, 2009, <http://yro.slashdot.org/story/09/07/17/2138213/Amazon-Pulls-Purchased-E-Book-Copies-of-1984-and-Animal-Farm>
- Google: Google Docs Online Help, Functions: Using GoogleLookup, , <http://docs.google.com/support/bin/answer.py?hl=en&answer=54199>
- heise: Kramer, Andre, Photoshop als Web-Service, 2007, <http://www.heise.de/newsticker/Photoshop-als-Web-Service--/meldung/86052>
- GW04: bono333, Accountzahlen von GW, , <http://galaxywarsforum.de/thread.php?postid=17752821#post17752821>
- Rapoza08: Jim Rapoza, RIA War Is Brewing, 2008, http://etech.eweek.com/content/application_development/ria_war_is_brewing.html
- Ihlenfeld07: Jens Ihlenfeld, Prism - Mozilla macht Websites zu Desktop-Applikationen, 2007, <http://www.golem.de/0710/55646.html>
- MozWiki09: Mozilla Wiki Users, Prism, , <https://wiki.mozilla.org/index.php?title=Prism&oldid=155644>
- Prism: Prism-Autoren, Features: Popup alerts, , <http://prism.mozilla.com/features/>
- MozWiki08: Mozilla-Wiki Users, Prism/FAQ, 2008, <https://wiki.mozilla.org/index.php?title=Prism/FAQ&oldid=99206>
- OOS: iCUBE Network Solutions, Online Operating System , 2007, <http://www.icube.at/oos.jsp>
- Tanenbaum03: Andrew S. Tanenbaum, Computernetzwerke, 2003
- Wilson08: Wilson, Chris, Compatibility and IE8, 2008, <http://blogs.msdn.com/ie/archive/2008/01/21/compatibility-and-ie8.aspx>
- Schonfeld08: Erick Schonfeld, Once Nearly Invisible To Search Engines, Flash Files Can Now Be Found And Indexed, 2008, <http://www.techcrunch.com/2008/06/30/once-nearly-invisible-to-search-engines-flash-files-can-now-be-found-and-indexed/>
- FlashCookies: Soltani et al, Flash Cookies and Privacy , 2009, <http://ssrn.com/abstract=1446862>
- Schmidt09: Schmidt, Thorsten, Ausgeflasht: Ein Versuch mit Gnash und Swfdec, 2009, http://www.freiesmagazin.de/mobil/freiesMagazin-2009-05-bilder.html#09_05_swfdec-gnash
- whatwg1: , , http://wiki.whatwg.org/wiki/FAQ#What_is_the_WHATWG.3F
- Meyer09: Meyer, David, HTML 5 drops open source video codec, , <http://www.zdnetasia.com/news/software/0,39044164,62055739,00.htm>
- WHATWG07: Ian Hickson et. al., , 2007, <http://www.whatwg.org/specs/web-apps/2007-10-26/multipage/section-video.html#expose>
- SVG: Anne van Kesteren, , 2009, <http://www.w3.org/TR/2009/WD-html5-diff-20090825/#mathml-svg>
- W3C09: Anne van Kesteren, HTML 5 differences from HTML 4, 2009, <http://www.w3.org/TR/2009/WD-html5-diff-20090825/>
- IPTC: IPTC International Press Telecommunications Council, The Information Interchange Model, , <http://www.iptc.org/cms/site/index.html?channel=CH0108>
- Riecks: David Riecks, The IPTC-NAA standards, , http://www.controlledvocabulary.com/imagetdatabases/iptc_naa.html
- Riecks2: David Riecks, , , <http://www.controlledvocabulary.com/index.html>
- DC08: The Dublin Core Metadata Initiative, Dublin Core Metadata Element Set, Version 1.1, 2008, <http://dublincore.org/documents/2008/01/14/dces/>
- WP09: Wikipedia contributors, Paris (disambiguation) , 2009, [http://en.wikipedia.org/w/index.php?title=Paris_\(disambiguation\)&oldid=314868549](http://en.wikipedia.org/w/index.php?title=Paris_(disambiguation)&oldid=314868549)
- geonames: The GeoNames Team, GeoNames Ontology, , <http://www.geonames.org/ontology/>
- Hesse02: Wolfgang Hesse, Informatiklexikon: Ontologie(n), 2002, http://www.gi-ev.de/no_cache/service/informatiklexikon/informatiklexikon-detailansicht/meldung/ontologien-57/
- W3C08: RDF-in-HTML Task Force, RDFa Primer -- Bridging the Human and Data Webs, 2008, <http://www.w3.org/TR/2008/NOTE-xhtml-rdfa-primer-20081014/>

çelik04: tantek çelik, kevin marks, real world semantics, 2004,
<http://www.tantek.com/presentations/2004etech/realworldsemanticspres.html>
MFwiki: Tantek Çelik, hCard examples, , <http://microformats.org/wiki/hcard-examples>
Birbeck08: Mark Birbeck, Microformats and RDFa are not as far apart as people think, 2008, <http://internet-apps.blogspot.com/2008/06/microformats-and-rdfa-are-not-as-far.html>
BBC: Michael Smethurst, Removing Microformats from [bbc.co.uk/programmes](http://www.bbc.co.uk/programmes), 2008,
http://www.bbc.co.uk/blogs/radiolabs/2008/06/removing_microformats_from_bbc.shtml
Prodromou08: Evan Prodromou, RDFa vs microformats, 2008,
http://evan.prodromou.name/RDFa_vs_microformats
W3C98: Tim Berners-Lee, Notation 3 Resources , , <http://www.w3.org/DesignIssues/N3Resources>
OWL2: Pascal Hitzler et al, OWL 2 Web Ontology LanguagePrimer -- W3C Working Draft 11 June 2009, ,
<http://www.w3.org/TR/2009/WD-owl2-primer-20090611/>
nvidia: NVIDIA Corporation, CUDA Zone -- The resource for CUDA developers, ,
http://www.nvidia.com/object/cuda_home.html
Welchering09: Peter Welchering, "Dunkle Netze" für jedermann, 2009,
<http://www.dradio.de/dlf/sendungen/computer/1009162/>
Zappi07: "Zappis Welt"-Blog, SPAM Statistik, 2007, <http://zappi.wordpress.com/2007/09/12/spam-statistik/>

Alle URLs zuletzt geprüft am 19.09.2009, Abends