

***Diplomarbeit***  
***zur Erlangung des akademischen Grades***  
***Magister rerum socialium oeconomicarumque***

**Thema der Diplomarbeit:**  
**Eine Ajax Literaturdatenbank**  
**mit ADOweb**

<b>Autor:</b>	<b>Gottfried Nindl</b>
<b>Matrikelnummer:</b>	0027663
<b>Email:</b>	<a href="mailto:e0027663@mail.student.tuwien.ac.at">e0027663@mail.student.tuwien.ac.at</a>
<b>Studienrichtung:</b>	Wirtschaftsinformatik
<b>Studienkennzahl:</b>	E 175
<b>Stammuniversität:</b>	Technische Universität Wien
<b>Betreuer:</b>	O. Univ. Prof. Dr. Dimitris Karagiannis
<b>Institut:</b>	Institut für Knowledge und Business Engineering
<b>Fakultät:</b>	Fakultät für Informatik
<b>Universität:</b>	Universität Wien
<b>Eingereicht am:</b>	1. April 2008

# Inhaltsverzeichnis

<b>Vorwort</b> .....	<b>IV</b>
<b>Eidesstattliche Erklärung</b> .....	<b>V</b>
<b>Zusammenfassung</b> .....	<b>VI</b>
<b>Abstract</b> .....	<b>VII</b>
<b>Abkürzungsverzeichnis</b> .....	<b>VIII</b>
<b>Abbildungsverzeichnis</b> .....	<b>X</b>
<b>Tabellenverzeichnis</b> .....	<b>XII</b>
<b>1 Einleitung</b> .....	<b>1</b>
1.1 Problemstellung .....	2
1.2 Zielsetzung .....	4
1.3 Vorgehensweise .....	6
<b>2 Technologische Grundlagen</b> .....	<b>8</b>
2.1 Verteilte Systeme als globales Modell .....	9
2.1.1 Kommunikationsmodell im Internet .....	10
2.1.2 Surfen im World Wide Web (WWW) .....	13
2.2 Webapplikationen und -services .....	16
2.2.1 Client und Server .....	16
2.2.2 Peer-to-Peer (P2P) .....	18
2.2.3 Webservice .....	19
2.2.4 Rich Internet Applikationen (RIA) .....	20
2.3 ADOweb .....	21
2.3.1 Positionierung .....	22
2.3.2 Architektur .....	24
2.3.3 Web 2.0 mit PROMOTE .....	26
2.4 Web 2.0 .....	28
2.4.1 Unternehmen im Wandel .....	30
2.4.2 Blogging .....	31
2.4.3 Social Software .....	33
2.4.4 Die Zukunft des Web .....	34
<b>3 Ajax</b> .....	<b>37</b>
3.1 Das Konzept .....	38
3.2 Übersicht und Vergleich Ajax Frameworks .....	42
3.2.1 Prototype .....	46
3.2.2 Script.aculo.us .....	48
3.2.3 Dojo .....	50
3.2.4 DWR .....	51
3.2.5 jQuery .....	53
3.2.6 Yahoo! UI (YUI) .....	54
3.2.7 Rico .....	56
3.2.8 ASP.NET Ajax .....	57
3.2.9 Xajax .....	59

3.2.10	GWT .....	61
3.2.11	ThinWire.....	63
3.2.12	Open-jACOB .....	64
3.2.13	Qooxdoo .....	65
3.2.14	RAP .....	67
3.3	Ergebnisse der Analyse .....	69
<b>4</b>	<b>Die Implementierung.....</b>	<b>74</b>
4.1	Zielsetzung .....	75
4.2	Architektur.....	77
4.2.1	Software Bibliotheken .....	78
4.2.2	Modell Repository .....	80
4.2.3	Klassendiagramme.....	83
4.3	Die Workbench.....	87
4.3.1	Technologischer Hintergrund .....	88
4.3.2	Modell Editor und Annotation.....	89
4.3.3	Suchmaschine .....	95
4.4	Einschränkungen und Probleme .....	97
4.5	Exkurs: Mouse Listener.....	99
<b>5</b>	<b>Schlusswort .....</b>	<b>108</b>
<b>6</b>	<b>Literaturverzeichnis .....</b>	<b>112</b>

## Vorwort

Das Internet ist eine kreative Werkstatt des menschlichen Geistes. Wikis, Foren oder Blogs sind Ausdrucksformen eines aktiven Verstandes, der für offenen Wissens- und Informationsaustausch empfänglich ist. Die Einsicht, dass das vollständige Erfassen von Zusammenhängen und Funktionsweisen der äußeren Welt nur durch kollektive Wissensarbeit erreicht werden kann, ist wahrscheinlich einer der wertvollsten Erkenntnisse unserer Zeit. Es bleibt den Lesern überlassen zu entscheiden, ob der entwickelte Prototyp auf diesem Gedanken gründet oder von anderen Motiven geprägt ist.

An dieser Stelle möchte ich mich bei allen bedanken, die direkt oder indirekt an der Entstehung dieser Diplomarbeit beteiligt waren. Mein erster Dank gebührt meinen Eltern, die mir die finanzielle Grundlage für mein Studium geschaffen haben. Dies ist vor allem auf meine Mutter zurückzuführen, weil sie tagtäglich hart und fleißig im Hotelbetrieb gearbeitet hat. Darüber hinaus hat mir mein Vater während meines Studiums wertvolle Praktika in England, den USA, sowie in Bangalore bei Siemens Indien vermittelt. Die Einblicke in andere Arbeitskulturen und Welten bescherte mir unbezahlbare Erfahrungen, die ich in Relation zu meiner Ausbildung setzen konnte. Vor allem danke ich meiner wundervollen Freundin Anja. Neben ihrer seelischen Unterstützung glänzte sie durch ihre präzisen Korrekturarbeiten und Verbesserungsvorschläge, die maßgeblich zur Qualität und zum Stil des Textdokuments beigetragen haben. Des Weiteren danke ich dem Leiter der BOC Forschungsabteilung Dr. Robert Woitsch, der das Projekt fachlich ausgezeichnet betreut und zum Abschluss die Diplomarbeit korrigiert hat. Meinem Professor Dimitris Karagiannis gebührt natürlich besonderer Dank, da er mir das Umfeld zur Entfaltung meiner Arbeitstätigkeit bereit gestellt hat. Die weiteren Mitarbeiter der BOC Forschungsabteilung, namentlich Mag. Vedran Blazevic, Mag. Wilfrid Utz, Mag.<sup>a</sup> Andrea Leutgeb, Mag. Vedran Hrgovic, Mag. Hannes Eichner und Mag. Bernhard Teuchmann, haben mich auch in Ausführung meiner Arbeit sehr unterstützt.

Die Ergebnisse dieser Diplomarbeit sind von meinem Standpunkt aus betrachtet sehr zufrieden stellend. Wissen über SWT, Ajax, ADOweb oder Web 2.0 war bei Arbeitsantritt so gut wie nicht vorhanden. Es konnte ein grundlegendes Verständnis über die Materie erworben werden. Ich kann nur hoffen, dass die Resultate auch für weitere Projekte bzw. Arbeiten dieser Art interessant, nützlich und einsetzbar sein werden.

## **Eidesstattliche Erklärung**

Hiermit versichere ich, die vorliegende Diplomarbeit ohne Hilfe Dritter und nur mir den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

---

Unterschrift des Autors (Gottfried Nindl)

## Zusammenfassung

Nach dem „Platzen der Dot-Com Seifenblase“ muss sich die Internet Branche auf neue Konzepte besinnen. Das Schlagwort Web 2.0 beschreibt einen neuen, kollaborativen Ansatz, der Benutzer zunehmend in den Gestaltungsprozess von Software involviert. Am Musterbeispiel der Online-Enzyklopädie Wikipedia kann man erkennen, dass kollektive Wissensarbeit zu einer Multiplikation des individuellen Wissens führt. Zur Etablierung eines solchen sozialen Umfelds wird ein Prototyp für eine webbasierte Literaturdatenbank entwickelt, welche ein Dokumenteninformationsportal für das Wissensmanagement einer Community bereit stellen soll. Das Ziel ist die Schaffung eines synergetischen, wachsenden und selbstregulierenden Informationsraums, in dem sich dessen Teilnehmer gegenseitig fördern und helfen können. Dafür gewährleistet die Technologie Ajax (Aynchronous JavaScript and XML) den Komfort und die Benutzerfreundlichkeit einer Rich Internet Applikation (RIA). Die Wissensmanagementmethode PROMOTE, welche als Anwendungsbibliothek des Modellierungswerkzeugs ADONIS implementiert ist, definiert die Bedeutung der abgelegten Dokumente. Das funktionale Verbinden von Ajax und ADONIS wird mit dem Webframework ADOweb bewerkstelligt.

Weiterhin beschreibt das Schlagwort Web 2.0 nichts anderes als die Metamorphose einer Benutzeroberflächentechnologie zu einer Plattform. Blogs und Wikis sind nur die ersten Vorboten einer neuen Generation von Applikationen, die kollaborative Umgebungen auf der Grundlage des Internets konstruieren. Das Web von morgen besitzt das Potential die Infrastruktur der virtuellen Realität bilden zu können, da auf viele Funktionen, wie z. B. die Partizipation und die Vernetzung, zurück gegriffen werden kann. Ajax wird in naher Zukunft großer Erfolg verheißen, da browserbasierte Applikationen den Betriebssystemdistributionen Konkurrenz machen werden. Diese Behauptung wird durch den entwickelten Prototyp untermauert, der auf dem ausgewählten Ajax Framework RAP eine RIA aufsetzt. Open Source Projekte können auch für kommerzielle Unternehmen interessant werden, da sie zum Anwerben neuer Kunden verwendet werden können. Die Etablierung einer Community ist entscheidend, wobei auch das Softwareunternehmen BOC mit ihrem Forschungsprojekt ADOweb ungeahnte Potentiale nutzen kann.

## Abstract

After the „burst of the dot-com bubble“, the Internet business has to evaluate new concepts. The keyword Web 2.0 describes a new collaborative approach to involve users in the design process of software. The Online-Encyclopaedia Wikipedia gives evidence to the fact that collective knowledge work results into a multiplication of individual knowledge. Therefore a prototype of a web-based literature database is developed in order to establish a social environment. This is realized by a document information portal that serves the needs of community knowledge management. The aim is to create a synergetic, growing and self-regulating information space where participants are able to subside and help each other. For this reason the technology Ajax (Aynchronous JavaScript and XML) provides the comfort and user-friendliness of a Rich Internet Application (RIA). The knowledge management method PROMOTE, which is implemented as an application library of ADONIS, defines the semantics of the archived documents. The functional coupling of Ajax and ADONIS is done by the web framework ADOweb.

Furthermore Web 2.0 describes the metamorphosis of a user interface technology to a platform. Blogs and Wikis are only the first pioneers of a new generation of applications which create collaborative environments based on the Internet. Tommorrow's web will potentially provide the infrastructure of a virtual reality, because it relies on many functions, like participations and networking. Promising prospective success, Ajax creates browser-based applications which will compete with distributions on operating systems. The developed prototype, which builds a RIA on top of the Ajax framework RAP, proves this. Open source projects might become interesting for commercial companies because they can be used to drum up new customers. The foundation of a community is crucial whereby the software company BOC with its research project ADOweb can exploit unknown potentials.

## Abkürzungsverzeichnis

ACM.....	Association for Computing Machinery
AJAX .....	Asynchronous JavaScript and XML
ASP.....	Active Server Pages
API.....	Advanced Program Interface
BOC.....	Business Objectives Company
CEO.....	Chief Executive Officer
CD.....	Compact Disk
CGI.....	Common Gateway Interface
CNN.....	Cable News Network
CORBA.....	Common Object Request Broker Architecture
CRM.....	Customer Relationship Management
CSS.....	Cascading Style Sheets
CVS.....	Concurrent Versioning System
DOM.....	Document Object Model
DBMS.....	Database Management System
DHTML.....	Dynamic HTML
DNS.....	Domain Name Service
DOM.....	Document Object Model
DWR.....	Direct Web Remoting
EPL.....	Eclipse Public License
FAQ.....	Frequently Asked Questions
FDDI.....	Fiber Distributed Data Interface
FTP.....	File Transfer Protocol
GUI.....	Graphical User Interface
GWT.....	Google Web Toolkit
HTML.....	Hyper Text Markup Language
HTTP.....	Hyper Text Transfer Protocol
IBM.....	International Business Machines
ICQ.....	I seek you
IE.....	Internet Explorer
IEEE.....	Institute of Electrical and Electronics Engineers
IIS.....	Internet Information Server
IM.....	Instant Messaging
IP.....	Internet Protocol
IPO.....	Initial Public Offering
IRC.....	Internet Relay Chat
ISAPI.....	Internet Server Application Programming Interface
JSF.....	Java Server Faces
JSNI.....	JavaScript Native Interface
JXTA.....	Juxtapose
MIME.....	Multipurpose Internet Mail Extensions
MS.....	Microsoft
MSN.....	Microsoft Network
MVC.....	Model-View-Control
NASDAQ.....	National Association of Securities Dealers Automated Quotations
NBC.....	National Broadcasting Company
OS.....	Operation System
OSI.....	Open Systems Interconnection
OSPF.....	Open Shortest Path First



OWL.....	Web Ontology Language
P2P.....	Peer-to-Peer
PC.....	Personal Computer
PDA.....	Personal Digital Assistant
PDF.....	Portable Document Format
PHP.....	PHP: Hypertext Preprocessor
RAD.....	Rich Application Development
RAP.....	Rich Ajax Platform
RCP.....	Rich Client Platform
RDF.....	Resource Description Framework
RIA.....	Rich Internet Application
RIP.....	Routing Information Protocol
RP.....	Rheinländische Post
RPC.....	Remote Procedure Call
RSS.....	Really Simple Syndication
RWT.....	Rich Widget Toolkit
SMTP.....	Simple Mail Transfer Protocol
SOAP.....	Simple Object Access Protocol
SPX.....	Sequenced Packet Exchange
SQL.....	Standard Query Language
SVG.....	Scalable Vector Graphics
SWT.....	Standard Widget Toolkit
UDDI.....	Universal Description, Discovery and Integration
UDP.....	User Datagram Protocol
UI.....	User Interface
UML.....	Unified Modeling Language
URI.....	Uniform Resource Identifier
URL.....	Uniform Resource Locator
WAR.....	Web Application Archive
W3C.....	World Wide Web Consortium
WSDL.....	Web Service Description Language
WWW.....	World Wide Web
XHTML.....	Extensible Hypertext Markup Language
XSLT.....	Extensible Stylesheet Language Transformation
XML.....	Extensible Markup Language
XUL.....	XML User Interface Language
YUI.....	Yahoo User Interface

## Abbildungsverzeichnis

Abbildung 1: Modifiziertes OSI Kommunikationsmodell im Internet .....	11
Abbildung 2: Kommunikation über verschiedene Netzwerke .....	12
Abbildung 3: Browserfenster mit HTML Seite .....	15
Abbildung 4: Architektur webbasierter Applikationen .....	17
Abbildung 5: Zwei-Weg-Webservice.....	19
Abbildung 6: Positionierung von ADOweb in die ADO-Produktfamilie .....	23
Abbildung 7: ADOweb Framework .....	25
Abbildung 8: Mind Map Social Software .....	34
Abbildung 9: Herkömmliche vs. Ajax Webapplikation .....	39
Abbildung 10: Umfrage über Ajax Frameworks und Plattformen .....	44
Abbildung 11: Prototype Architektur mit externem Service.....	47
Abbildung 12: Prototype Postleitzahl Überprüfung .....	47
Abbildung 13: Script.aculo.us Architektur.....	49
Abbildung 14: Script.aculo.us Drag und Drop Demo .....	49
Abbildung 15: Dojo Fisheye Demo mit skalierbaren SVG Grafiken.....	50
Abbildung 16: Dojo/DWR Architektur .....	51
Abbildung 17: DWR Tabellen Editor.....	52
Abbildung 18: jQuery HTML Editor .....	53
Abbildung 19: Yahoo Architektur .....	55
Abbildung 20: Yahoo! Text Editor.....	55
Abbildung 21: Rico Architektur .....	56
Abbildung 22: Rico Wetter .....	57
Abbildung 23: ASP.NET Architektur .....	58
Abbildung 24: ASP.NET Slideshow und Toolkit .....	59
Abbildung 25: Xajax Architektur .....	60
Abbildung 26: Xajax Graphity Wall .....	60
Abbildung 27: GWT Architektur .....	62
Abbildung 28: Google Mail.....	62
Abbildung 29: ThinWire Architektur .....	63
Abbildung 30: ThinWire CRM System.....	64
Abbildung 31: Draw2D von Open-jACOB .....	65
Abbildung 32: Qooxdoo Architektur.....	66
Abbildung 33: Qooxdoo Test Runner .....	66
Abbildung 34: RAP Architektur.....	68
Abbildung 35: RAP Workbench .....	68
Abbildung 36: Kombination und Auswahl von Frameworks.....	71
Abbildung 37: Anwendungsprozess der Literaturdatenbank .....	75
Abbildung 38: Einbettung der Literaturdatenbank in das ADOweb Service Framework.....	77
Abbildung 39: Software Bibliotheken der Ajax Literaturdatenbank .....	79
Abbildung 40: Lose Dokumente in einem Wissensressourcen Pool.....	80
Abbildung 41: Strukturiertes Dokumentarchiv mit Wissensquellen.....	81
Abbildung 42: Referenzierte Dokumente eines Topics.....	82
Abbildung 43: Ausschnitt einer Topic Map zur Annotation von Dokumenten .....	82
Abbildung 44: Basispaket (com.boc_eu.adoweb.component.litdb) .....	83
Abbildung 45: AdoScript Objects (com.boc_eu.adoweb.component.litdb.aso) .....	84
Abbildung 46: RAP Benutzeroberfläche (com.boc_eu.adoweb.component.litdb.ui) .....	86
Abbildung 47: Modell Editor & Annotation Perspektive.....	90
Abbildung 48: Explorer View des Literatur Repository .....	91
Abbildung 49: Neues Dokument anlegen.....	91

Abbildung 50: Detailansicht eines Dokuments .....	92
Abbildung 51: Themenlandkarte in der Annotation View .....	93
Abbildung 52: Blog View .....	94
Abbildung 53: Neuer Blog Eintrag.....	94
Abbildung 54: Literatur Inbox View .....	94
Abbildung 55: Perspektive der Suchmaschine .....	95
Abbildung 56: Suchparameter View .....	96
Abbildung 57: Suchresultate View .....	96

## Tabellenverzeichnis

Tabelle 1: Zwecke von Netzwerken laut Zara.....	3
Tabelle 2: Wichtige ADOweb Komponenten .....	26
Tabelle 3: Konzeptioneller Unterschied Web 1.0 zu Web 2.0 .....	28
Tabelle 4: Ajax Framework Vergleich .....	70
Tabelle 5: Verwendete PROMOTE Modelltypen .....	80

# 1 Einleitung

Am 10. März 2000 erreichte der NASDAQ Composite Aktienindex, welcher 3.000 Unternehmen aus dem Technologiesektor, wie u. a. Microsoft oder Dell, umfasst, einen Höchststand von 5.048,62 Punkten und verdoppelte damit seinen Wert innerhalb von einem Jahr.<sup>1</sup> Dieses Hoch währte jedoch nicht lange, denn auf den ruckartigen Aufschwung folgte eine ebenso schnelle Talfahrt. Dieses Ereignis wird als das „Platzen der Dot-Com Seifenblase“ bezeichnet, weil es vor allem von Unternehmen der „New Economy“ verursacht wurde, die die verheißungsvollen Prognosen aus der Zeit des Börsengangs verfehlten.<sup>2</sup> Das Misstrauen wurde besonders durch das Kursieren von Gerüchten über eine schlechte Liquidität hoch bewerteter Dot-Com Firmen geschürt. Diese Umstände veranlassten viele Anleger<sup>3</sup> ihr Kapital vom Aktienmarkt abzuziehen. Panische Verkäufe führten zu einem signifikanten Kursverfall, der den NASDAQ Composite Aktienindex bis zum Oktober 2002 auf etwas mehr als ein Fünftel des Höchststandes zusammen schrumpfen ließ (1.114,11 Punkte).<sup>4</sup> Die Liste der Bankrotterklärungen ist lang, wobei beträchtliche Geldbeträge pulverisiert wurden. Ein großer Reinfluss war u. a. der elektronische Lebensmittelhandel Webvan.com, der einen IPO von \$ 375 Mio. aufbrachte und zeitweilig eine Spitze von \$ 1,2 Mrd. erreichte. Die Firma beendete die Geschäftstätigkeit nach nur zwei Jahren und entließ 2.000 Angestellte, weil die Umsätze nicht die dünnen Gewinnspannen im Lebensmittelgeschäft decken konnten.<sup>5</sup> Pets.com mit einem IPO von \$ 82,5 Mio. bot die einmalige Möglichkeit der Lieferung von Katzenstreu an eine beliebige Adresse. Das Angebot stieß bei den Tierhaltern nur auf mäßige Begeisterung, weshalb die Waren aus Werbegründen mit Verlust verkauft wurden. Die Geschäftsstrategie führte nach nur neun Monaten zur Zahlungsunfähigkeit des Unternehmens.<sup>6</sup> Insgesamt war die dramatische Krise vor allem eine Konsequenz zu hoher Erwartungen, die in fragwürdige bzw. überschätzte Vorhaben gesetzt wurden. Viele Investoren wurden durch das Zusammenbrechen des neuen Marktes zu Recht abgeschreckt. Originelle Ideen, zusammen mit klugen Geschäftsmodellen und realistischen Finanzierungsplänen waren von Nöten, um das Vertrauen der Kapitalgeber zurück zu gewinnen.

---

<sup>1</sup> Vgl. EDGAR Online (2008): „NASDAQ Composite IXIC“

<sup>2</sup> Vgl. Frenz, Clements (2003): „Die Chronik einer Kapitalvernichtung“

<sup>3</sup> Aus Lesbarkeitsgründen verzichtet der Autor auf die gesetzlich vorgeschriebene geschlechtliche Gleichstellung. Die weiblichen Personen, wie in diesem Fall die Anlegerinnen, sind auch gemeint.

<sup>4</sup> Vgl. EDGAR Online (2008): Ebenda

<sup>5</sup> Vgl. German, Kent (2008): „Top ten dot-com flops“

<sup>6</sup> Vgl. German, Kent (2008): Ebenda

## 1.1 Problemstellung

In der Zeit nach dem „Platzen der Dot-Com Seifenblase“ wurde in vielen überlebenden Unternehmen eine gründliche Revision durchgeführt, um nicht einem sich wiederholenden Szenario zum Opfer zu fallen. Das Platzen von Aktienmarktblasen ist nämlich alle 13 Jahre zu beobachten, wobei die realwirtschaftlichen Auswirkungen ca. 2 ½ Jahre andauern.<sup>7</sup> Ein Beispiel für das Überstehen dieser schweren Zeit ist die Suchmaschine Google, dessen durchschnittlicher Umsatz sich in den letzten fünf Jahren ver Hundertfachte und mit einem Marktwert von \$ 156 Mrd. (Stand: 22. November 2006) dotiert wird.<sup>8</sup> Damit übertrifft Google renommierte Riesen der Geschäftswelt wie u. a. Coca-Cola, IBM, Cisco oder Time Warner.<sup>9</sup> Als Hauptgründe für den Erfolg der derzeit dominierenden Internetfirmen werden das Geschäftsmodell, freie Services anstatt Softwarepaketlizenzen, offene Programmierschnittstellen (API) und der Gestaltungsfreiraum kollektiver Intelligenz, wie sie z. B. bei Youtube, Flickr, Amazon, Ebay oder Google's PageRank der Fall sind, genannt.<sup>10</sup> Die Vermeidung von Assoziationen mit den untergegangenen Dot-Com Firmen ist den heute präsenten Marktführern ein wichtiges Anliegen, da sie ihre unterscheidenden Vorzüge nur allzu gern betonen. Es stellt sich die Frage, ob es sich dabei nur um eine Werbelinie handelt, welche alte Produkte in neuen Verpackungen zu verkaufen beabsichtigt, oder ob sich die Geschäftsstrategien wirklich substantiell unterscheiden.

Schlagwörter waren immer schon ein beliebtes Mittel, um Ideen und Philosophien zu verbreiten. Ein wichtiger Begriff im Zusammenhang mit dem WWW ist das Semantic Web, dessen Einführung gegen Ende der 90er Jahre auf das Engagement des Webbrowser-Erfinders Tim Berners-Lee zurück zu führen ist.<sup>11</sup> Das Scheitern vieler Dot-Com Firmen und das anschließende Suchen nach neuen Lösungswegen verschaffte diesem technologiebezogenen Konzept mehr Aufmerksamkeit. Das Ziel lag<sup>12</sup> dabei in der Steigerung der Informationsqualität von Webinhalten, die u. a. mit der Metadatenauszeichnungssprache RDF<sup>13</sup> und der Ontologiesprache OWL<sup>14</sup> verfolgt wird, um die Interoperabilität zwischen Maschinen bzw. Applikationen zu verbessern. Parallel dazu entwickelten sich kollaborative

---

<sup>7</sup> Vgl. Deutsche Bank Research (2003): „Dot-Com Crash: Talsohle in Deutschland durchschritten“, S. 13

<sup>8</sup> Vgl. Otte, Prof. Dr. Max (2006): „GOOBAYHOO!“

<sup>9</sup> Vgl. BlinkBits.com (2007): „Google Wikipedia Rss Feed“

<sup>10</sup> Vgl. Holz, Patrick (2006): „Nutzung kollektiver Intelligenz“

<sup>11</sup> Vgl. Berners-Lee, Tim (1999): „Transcript of Tim Berners-Lee's talk to the LCS 35th Anniversary celebrations“

<sup>12</sup> und liegt auch heute noch

<sup>13</sup> W3C (2004): „RDF Resource Description Framework“

<sup>14</sup> W3C (2004): „OWL Web Ontology Language“

Technologien, bei denen die gemeinschaftliche Arbeit zugunsten intellektueller Bemühungen<sup>15</sup> bzw. der soziale Austausch der teilnehmenden Personen im Vordergrund steht. Netzwerke können laut Zara<sup>16</sup> drei Zwecken dienen, wobei eine kollaborative Ausrichtung die Benutzer am meisten bevollmächtigt:

<i>Zweck</i>	<i>Rolle der Benutzer</i>	<i>Beispiele</i>
Information	Empfänger	Suchmaschinen, File Sharing, FAQ
Kommunikation	Sender und Empfänger	Email, Chat, Wiki, E-Learning
Kollaboration	Sender, Empfänger und Agent	Diskussionsforum, Blog, E-Coaching

**Tabelle 1: Zwecke von Netzwerken laut Zara**

Fortgeschrittene, kollaborative Netzwerke gehen weit über Kommunikation und Information hinaus, weil sie Regeln der Interaktion festlegen, die zur Kategorisierung und Evaluierung von Information genutzt werden können.<sup>17</sup> Der Erfolg der Online-Enzyklopädie Wikipedia<sup>18</sup> oder dem File Sharing System Bittorrent<sup>19</sup>, die sich mit kostenlosen Dienstleistungen und der Unterstützung vieler Freiwilliger schnell verbreitet haben, überraschte nicht nur Experten. Der Verlag O'Reilly und der Konferenz- bzw. Messeveranstalter MediaLive International versuchten daraufhin, die Ursachen dieser gesellschaftlichen Massenphänomene zu analysieren. Die Ergebnisse des Brainstormings dazu wurden im Artikel „What is Web 2.0“<sup>20</sup> veröffentlicht. Darin werden Unterschiede zum herkömmlichen WWW, wie z. B. Veröffentlichung vs. Beteiligung, identifiziert, um zukünftige Entwicklungen, wie z. B. das Web als Plattform, abzuleiten. Im gleichen Jahr, in dem der O'Reilly Artikel erschien, verfasste Jesse Garret von Adaptive Paths einen Aufsatz mit dem Titel „Ajax: A New Approach to Web Applications“.<sup>21</sup> Dort wird die Architektur einer Technologie skizziert, die das Ausführen von desktop-ähnlichen Applikationen auf dem Webbrowser erlaubt. Darauf folgten eine Reihe von Entwicklungsbibliotheken, zunächst in JavaScript, später in PHP, Java etc., die Funktionen zur Implementierung von Rich Internet Applikationen (RIA) bereit stellen. Die Standpunkte über die Mächtigkeit der Technologie sind umstritten und lassen sich mit zwei polarisierenden Fragen zusammenfassen: Sind Textverarbeitungsprogramme, wie

<sup>15</sup> Vgl. Encyclopædia Britannica Online (2008): „Collaboration“

<sup>16</sup> Vgl. Zara, Olivier (2004): „Managing Collective Intelligence, Toward a New Corporate Governance“ , S. 43

<sup>17</sup> Vgl. Zara, Olivier (2004): Ebenda, S. 44

<sup>18</sup> Vgl. Wikimedia Foundation Inc. (2007): „Wikipedia“

<sup>19</sup> Vgl. BitTorrent, Inc.(2007) „Bittorrent“

<sup>20</sup> Vgl. O'Reilly, Tim (2005): „What is Web 2.0“

<sup>21</sup> Vgl. Garret, Jesse (2005): „Ajax: A New Approach to Web Applications“

MS Word<sup>22</sup>, Bildbearbeitungswerkzeuge, wie Adobe Photoshop CS3<sup>23</sup>, oder Unternehmens- und Geschäftsmodellierungsapplikationen, wie ADONIS<sup>24</sup>, bald als Services über das Internet verfügbar? Oder handelt es sich bei Ajax doch nur um eine überschätzte Technologie, die herkömmliche Webseiten ein wenig mit netten Effekten aufbessert?

Bei Ajax und Web 2.0 handelt es sich um offene Definitionen, weshalb es nicht verwunderlich ist, dass die Meinungen über deren Umsetzungen sehr unterschiedlich sind. Die Konzepte und Möglichkeiten, die sich hinter diesen Begriffen verbergen, sollen mit Hilfe einer konkreten Implementierung, deren Zielsetzung im folgenden Abschnitt erläutert wird, verifiziert werden.

## **1.2 Zielsetzung**

Freie Dienstleistungen und der informelle Charakter von vielen Web 2.0 Portalen, wie z. B. dem Social Bookmarking Dienst del.icio.us<sup>25</sup> oder der Online-Enzyklopädie Wikipedia, tragen sehr zu deren Popularitätsgrad bei. Der stetige Wissenszuwachs auf diesen Webseiten und das uneigennützig Bereitstellen von Information durch die Teilnehmer ist eine erstaunliche Entwicklung, die sich manche wissenschaftliche Organisation und kommerzielle Firmen wünschen würde. Das Kernproblem liegt in der Identifizierung der richtigen Wissensflüsse, die Personen dazu veranlasst, Wissen einzustellen und zu teilen. Die Modellierung der Prozesse erfolgt daher mit der Wissensmanagementmethode PROMOTE<sup>26</sup>, welche das gewünschte Verhalten der Teilnehmer, die Arbeitsumgebung und die thematischen Inhalte abbildet. Die Aufgabe im Rahmen dieser Diplomarbeit ist die Entwicklung eines Prototyps für eine webbasierte Literaturdatenbank, welche einfache Informationsservices zur Verwaltung eines physischen Dokumentenarchivs bereit stellt. Die Mitglieder einer Arbeitsgruppe sollen einerseits fachrelevante Literatur schnell und effektiv suchen können und andererseits die Möglichkeit bzw. die Berechtigung zur Einstellung von neuen Dokumenten besitzen. Das Ziel ist die Schaffung eines organisch wachsenden Informationsraums, in dem sich dessen Teilnehmer gegenseitig fördern und helfen können.

---

<sup>22</sup> Vgl. Microsoft (2007): „MS Office Online“

<sup>23</sup> Vgl. Adobe (2007): „Photoshop CS3 Edition“

<sup>24</sup> Vgl. BOC (2007): „Prozessorientierte Anwendungsentwicklung mit ADONIS“, S. 3ff

<sup>25</sup> del.icio.us corporate (2007): „del.icio.us – social bookmarking“

<sup>26</sup> Vgl. Karagiannis, Dimitris und Telesko R. (2000): „The EU-Project PROMOTE: A Process-oriented Approach for Knowledge Management“



Den Anreiz zum Wissenstransfer sollen einerseits die Inhalte und andererseits die Benutzerfreundlichkeit der Plattform geben. Deswegen setzt der geplante Prototyp als Gestaltungswerkzeug der direkten Endbenutzerinteraktion die Technologie Ajax ein. Das Wissensmanagementprodukt PROMOTE<sup>®</sup> erfasst des Weiteren Eigenschaften, wie z. B. Name, Ort oder Besitzer und Metainformationen über Literatur, wie z. B. Kategorisierungen, Schlagwörter oder Themenzuordnungen ähnlich wie in den Fachbibliotheken IEEE<sup>27</sup>, ACM<sup>28</sup> oder scholar.google.com<sup>29</sup>. Die definierten Wissensdaten werden in Form von PROMOTE Modellen verwaltet. Dienste, die Informationen aus Modellen aufrufen und verändern können, werden vom Webframework ADOweb<sup>30</sup> bewerkstelligt. Unter diesen Voraussetzungen ergibt sich folgende Architektur für die Literaturdatenbank:

- *Webframework:* ADOweb mit Ajax Technologie
- *Produktlogik:* PROMOTE<sup>®</sup> als Wissensmanagementmethode
- *Modellhaltung:* ADONIS<sup>®</sup> als Modelldatenbank

Die Realisierung des Prototyps einer Literaturdatenbank soll auf einer Recherche aufbauen, die sich vor allem um die aktuellen Trends des WWW, Web 2.0 und aktuelle Webtechnologien dreht. Das Hauptinteresse bezieht sich dabei auf eine Analyse des Begriffs Ajax, dessen Konzept mit praktischen Umsetzungen exemplarisch veranschaulicht werden soll. Aufgrund der genannten Punkte ergeben sich folgende Fragestellungen, die für diese Diplomarbeit relevant sind:

- Was ist das Wesen von verteilten Systeme und des WWW?
- Was sind Webapplikationen, Rich Internet Applikationen und ADOweb?
- Was ist das Besondere der Web 2.0 Philosophie gegenüber dem traditionellen WWW?
- Was ist das Konzept von Ajax, und welche Frameworks bieten sich für die professionelle Entwicklung von Rich Internet Applikationen an?
- Wie können ADONIS<sup>®</sup>, PROMOTE<sup>®</sup>, ADOweb und Ajax in Form des Szenarios einer Literaturdatenbank in einem Prototypen implementiert werden?
- Welche Probleme und Einschränkungen traten bei der Umsetzung des Prototyps auf?

---

<sup>27</sup> Vgl. IEEE (2007): „The world's leading professional association for the advancement of technology”

<sup>28</sup> Vgl. ACM (2007): „Association for Computing Machinery”

<sup>29</sup> Vgl. Google Scholar (2007): „Scholar.google.com”

<sup>30</sup> Adoweb ist ein webbasiertes Forschungsprojekt der BOC, für Details siehe Kapitel 2.3

- Welche Erkenntnisse und Schlussfolgerungen können aus diesem Projekt gezogen werden und welchen Mehrwert hat eine Ajax Integration für bestehende Produkte?
- Welche zukünftigen Entwicklungen im Bereich des Webs sind absehbar?

Wie nun zur Beantwortung dieser Fragen vorgegangen wird und welche Gliederung sich daraus ergibt, beschreibt das nächste Kapitel über die Vorgehensweise näher.

### **1.3 Vorgehensweise**

Der Erwerb von detailliertem Wissen über verteilte Systeme, Architekturen von Webapplikationen bzw. Services, Server- und Clienttechnologien oder der Semantik von Inhalten legt die Grundlage für alle weitere Schritte. Diese Basis wurde zum Großteil schon im Verlauf des Wirtschaftsinformatikstudiums aufgebaut und soll im Zuge dieser Diplomarbeit weiter vertieft werden. Fundierte Kenntnisse über die Funktionsweise und Anwendung von aktuellen Webtechnologien, sowie fortgeschrittene Entwicklungserfahrung mit der Programmiersprache Java<sup>31</sup> sind zwingende Voraussetzungen für den Entwurf dieser webbasierten Literaturdatenbank. Ausgehend von diesem theoretischen Basiswissen werden Informationen über den aktuellen Entwicklungsstand von firmeninternen Programmen und Projekten der BOC<sup>32</sup> eingeholt. Dies beinhaltet Informationen über die Applikation ADONIS<sup>®</sup>, den zugrunde liegenden Datenbanken, dem darauf aufbauenden Wissensmanagementprodukt PROMOTE<sup>®</sup>, sowie den aktuellen Webservices von ADOWeb. Danach erfolgen eine Beschreibung des Konzepts, sowie einer Recherche ausgewählter Frameworks von Ajax. Damit soll ein Überblick über die Möglichkeiten dieser Technologie verschafft werden, um geeignete Werkzeuge für die Implementierung des Prototyps zu evaluieren. Im Zuge dessen werden Methoden des Web 2.0 ausfindig gemacht, um wichtige Eigenschaften dieses Ansatzes herauszuarbeiten. Das erworbene Wissen und die Werkzeuge werden im nächsten Schritt für das Konstruieren einer statischen Benutzeroberfläche (GUI) verwendet. Das GUI wird danach mit dem webbasierten Framework ADOWeb verbunden damit mit dem ADONIS<sup>®</sup> Modell Repository kommuniziert werden kann, um Werte dynamisch auszulesen bzw. schreiben zu können. Die abschließenden Testdurchläufe ermitteln die Stabilität und Performanz des Prototyps. Auftretende Fehler und Mängel sollen

---

<sup>31</sup> Sun Developer Network (2007): „Java Technology“

<sup>32</sup> BOC (2007): „Geschäftsprozess- und Unternehmensmodellierung“

in dieser Phase erkannt und behoben werden. Schlussendlich erfolgt die Auslieferung bzw. Übergabe des entwickelten Prototyps auf einer CD.

Zunächst werden dazu in Kapitel 2 die technologischen Grundlagen dieser wissenschaftlichen Arbeit beschrieben. Es beginnt mit einer Einführung in Netzwerke, Internet und WWW. Des Weiteren werden allgemeine Architekturen von Webapplikationen, sowie die spezielle Infrastruktur des ADOweb erläutert. Außerdem wird der Begriff Web 2.0 näher betrachtet. Hier sollen die Unterschiede zum herkömmlichen Web konzeptionell herausgearbeitet und anschließend mit Anwendungsbeispielen erläutert werden. Die Resultate der Recherche über Ajax Frameworks werden im dritten Kapitel präsentiert, um daraus eine Auswahl zu treffen, die für die Umsetzung der Ajax Literaturdatenbank als geeignet erscheinen. Das vierte Kapitel stellt den eigentlichen Prototypen vor. Die Architektur soll in diesem Abschnitt mit UML modelliert werden. Danach wird die Benutzersicht der Literaturdatenbank mit einigen Screenshots vorgestellt, die mit den wichtigsten Funktionalitäten der Applikation erläutert werden. Im Zuge dessen soll auf Probleme und Einschränkungen hingewiesen werden, die im Laufe der Entwicklung aufgetreten sind. Zur Abrundung werden exemplarisch ausgewählte Code-Fragmente besprochen, die speziell für die Anforderungen entwickelt werden mussten. Das letzte Kapitel fasst zum Abschluss die wichtigsten Erkenntnisse und Erfahrungen dieser Diplomarbeit zusammen und gewährt noch einen kurzen Ausblick über mögliche Entwicklungen des Internets.

## 2 Technologische Grundlagen

Die Themen dieses Kapitels geben einen Überblick der Konzepte und Begriffe von Netzwerktechnologien und -architekturen. Es wird hier nicht der Anspruch auf Vollständigkeit erhoben, sondern nur jene Aspekte herausgearbeitet, die für die Arbeit nützlich sind. Als Basis wird gewisses Fachwissen in den Bereichen Hardware, Software und Betriebssysteme vorausgesetzt. Des Weiteren ist es hilfreich, Grundkenntnisse über Browser, Server und Webtechnologien zu besitzen. Die folgenden Erklärungen stützen sich weitgehend auf Modelle, die schrittweise die softwareperspektivische Anatomie eines verteilten Informationssystems enthüllen. Dies soll genügend Hintergrundinformationen liefern, um die weiteren Kapitel und die Positionierung des begleitenden Projekts, einer Ajax Literaturdatenbank, besser zu verstehen.

Der erste Abschnitt beschreibt das Prinzip eines verteilten Systems anhand der konkreten Umsetzung des Internet. Zunächst wird der Nachrichtenaustausch in einem sechsschichtigen Kommunikationsmodell skizziert und von jeder einzelnen Ebene betrachtet. Dann wird als beliebteste Anwendung des Internets das dokumentenbasierte, verteilte System des World Wide Web (WWW) vorgestellt. Es sollen Merkmale, die zur allgemeinen Benutzbar- und Verständlichkeit beigetragen haben, erwähnt werden. Der zweite Abschnitt behandelt Entwurfsmöglichkeiten von Webapplikationen und -services. Diese erweitern das WWW mit Technologien, die in der Lage sind, anspruchsvollere Aufgabenstellungen zu realisieren. Danach wird das Prinzip eines Webservice erläutert, welcher flexibel in jede beliebige Umgebung eingebaut werden kann. Eine Beschreibung von Rich Internet Applikationen (RIA), die zusätzliche, generische Funktionalitäten auf der Benutzerseite betreiben, soll die Thematik abrunden. Der nächste Abschnitt befasst sich mit dem Portal ADOweb der BOC. Dieses hat zum Ziel, die Funktionalität der betriebssystembasierten Produkte, wie z. B. ADONIS<sup>®</sup> oder ADOscore<sup>®</sup>, über das Web verfügbar zu machen. Darüber hinaus wird das Portal mit der wissensbasierten Produktlogik PROMOTE<sup>®</sup> versehen, die die Semantik der Anwendung steuert. Zuletzt folgt ein kurzer Einblick in das Konzept des Web 2.0, um Schlüsselprinzipien zu identifizieren, die für den Anwendungsbereich einer Literaturdatenbank eingesetzt werden können.

## 2.1 Verteilte Systeme als globales Modell

Zwei Technologiefortschritte bauten das Fundament von verteilten Systemen.<sup>33</sup> Der erste war die Einführung von schnellen und zuverlässigen Computernetzwerken, die hauptsächlich aus amerikanischen Militär- und Universitätsprojekten, wie u. a. SAGE, APRANET, MilNet, USENET oder NSFnet, entstanden sind. 1969 zählte das APRANET noch vier bescheidene Knoten, die sich bis 1989 auf ca. 100.000 vermehren sollten. Zum Ende des Jahres 2007 überschritt die Zahl der Internetteilnehmer, so genannten Hosts, schon die 500 Mio. Grenze.<sup>34</sup> Eine in den nächsten Jahren dem Wachstum bedrohliche Einschränkung stellt lediglich das derzeitige verwendete Interprotokoll dar, welches „nur“ 32 Bit Adressen benutzt,<sup>35</sup> was bedeutet, dass das Kontingent auf etwas mehr als 4 Mrd. Hosts begrenzt ist. Ansonsten zeichnet diese Skalierfähigkeit das Internet aus, dessen Geheimnis nicht zuletzt in der Namensschöpfung verborgen liegt. „Internet“ heißt wörtlich übersetzt „Verbundnetz“ und leitet sich vom Englischen „**I**nterconnected“ (miteinander verbunden) und „**N**etworks“ (Netzwerke) ab.<sup>36</sup> Es besteht aus einem elektronischen Zusammenschluss vieler, verschiedener Netzwerke, die über gemeinsame Schnittstellen miteinander kommunizieren können. Die Verbreitung von Netzwerken hängt aber maßgeblich von der Anzahl der Teilnehmer ab. Diese Notwendigkeit wurde durch die zweite, parallel verlaufende Entwicklung von leistungsfähigen Mikroprozessoren Mitte der 80er Jahre ermöglicht. Sie wiesen die gleiche Rechenleistung wie die damals üblichen, großen Mainframes auf, konnten jedoch zu einem Bruchteil der Kosten produziert werden. Die Tatsache, dass handliche Endgeräte in kurzer Zeit zu erschwinglichen Preisen der mittelständischen Bevölkerung produziert werden können, ist ein äquivalent entscheidender Faktor der Globalisierung des Internets und wird nur allzu gerne vernachlässigt.

Die erwähnten Voraussetzungen bilden den technologischen Rahmen von verteilten Systemen. Das bedeutet jedoch nicht, dass jedes Netzwerk zwangsläufig ein verteiltes System sein muss, welches sich wiederum über viele Netzwerke erstrecken kann. Es handelt sich im eigentlichen Sinne um ein Paradigma, welches Richtlinien zum Verbinden von einander entfernter Geräte vorgibt. Das Wesen von verteilten Systemen lässt sich mit der folgenden Definition charakterisieren:

---

<sup>33</sup> Vgl. Tanenbaum, Andrew und Van Steen, Martin (2003): „Verteilte Systeme“, S. 17

<sup>34</sup> Vgl. ISC - Internet System Consortium (2007): „Number of Internet Hosts“

<sup>35</sup> Vgl. Postel, Jonathan (1981): „Internet Protocol“, S. 9

<sup>36</sup> Vgl. Postel, Jonathan (1981): Ebenda, S. 1

„Ein verteiltes System ist eine Menge voneinander unabhängiger Computer, die dem Benutzer wie ein einzelnes, kohärentes System erscheinen.“<sup>37</sup>

In einem verteilten System existieren also viele Benutzer, die z. B. mit Hilfe von PCs, Laptops, Mobiltelefonen, PDAs etc. interagieren. Die Software auf diesen Geräten ist im Gegensatz zu einfachen Terminals auch ohne eine Anbindung an einen Zentralrechner, wenn auch beschränkt, funktionsfähig. Zur Schaffung einer Kommunikationsbasis werden die Teilnehmer mit standardisierten Schnittstellen, welche physische Komponenten sowie logische Protokolle mit einschließt, ausgestattet, damit sie untereinander Nachrichten bzw. Signale austauschen können. So kann eine Schnittstelle von verschiedenen Benutzern zur gleichen Zeit beansprucht werden, was bei der Nutzung identer System- bzw. Speicherressourcen Konflikte oder gar einen Systemabsturz bewirkt.<sup>38</sup> Diese Problematik wird mit dem Prinzip der *Nebenläufigkeit* gelöst. Konkurrente Aufrufe von Schnittstellen werden als separate Prozesse behandelt, die scheinbar parallel und unabhängig von einander exekutieren. Auf Applikationsebene werden sie als Instanzen eines ausführenden Programms wahrgenommen, während sie dem Betriebssystem als Entitäten erscheinen, denen bestimmte Systemressourcen zugeordnet sind. Leichtgewichtige Prozesse, so genannte „Threads“, können darüber hinaus die gleichen Applikationsdatenstrukturen teilen, weil der Zugriff darauf mit Hilfe einer Synchronisierungsstrategie kontrolliert wird.<sup>39</sup> Die Nebenläufigkeit ist ein Aspekt, der die Transparenz von verteilten Systemen gewährleistet. Jeder Benutzer hat das Gefühl alleine und ausschließlich auf dem lokalen Gerät zu operieren, obwohl sich die benötigten Systemressourcen an verschiedenen, weit entfernten Orten befinden können, auf die auch andere Teilnehmer Zugriff haben. Wie diese Interaktion auf Basis von Kommunikationsprotokollen im Internet ermöglicht wird und wie das World Wide Web (WWW) darauf aufbaut, führen die beiden folgenden Unterabschnitte näher aus.

### **2.1.1 Kommunikationsmodell im Internet**

Die Erfindung des TCP (Transmission Control Protocol) Protokolls von Bob Kahn und Vinton G. Cerf im Jahre 1983 und die Kombination mit dem IP (Internet Protocol) von Jonathan Postel im Jahre 1981 wird heute als der De-facto-Standard der

---

<sup>37</sup> Tanenbaum, Andrew und Van Steen, Martin (2003): Ebenda, S. 18

<sup>38</sup> Vgl. Bovet, Daniel P. und Cesati, Marco (2005): „Understanding the Linux Kernel“, S. 79f

<sup>39</sup> Vgl. Bovet, Daniel P. und Cesati, Marco (2005): Ebenda, S. 79f

Netzwerkkommunikation betrachtet.<sup>40</sup> Manche gehen sogar schon soweit zu sagen: „Internet ist alles, was durch TCP/IP verbunden ist“<sup>41</sup>. In der vielfältigen Welt des Internets finden natürlich auch noch andere Protokolle, wie z. B. FTP, SMTP, UDP, SPX, RIP oder OSPF, Verwendung.<sup>42</sup> Bei TCP/IP handelt es sich um eine Verbindungslogik auf zwei Ebenen, nämlich der Vermittlungs- und Transportschicht, die im modifizierten OSI Schichtmodell des Internets auf der dritten bzw. vierten Ebene angesiedelt werden kann:

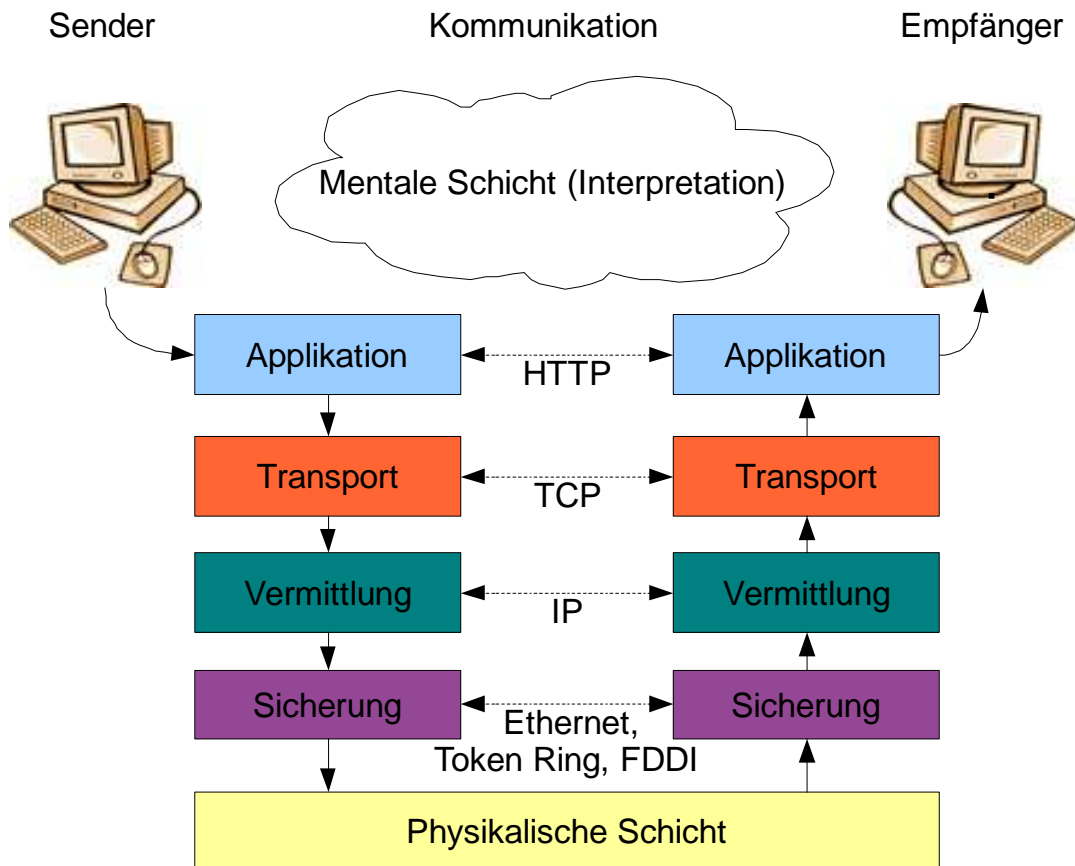


Abbildung 1: Modifiziertes OSI Kommunikationsmodell im Internet<sup>43</sup>

Die einzelnen Schichten haben den Zweck, dass bestimmte Aspekte der Kommunikation in verwaltbare Teile gegliedert werden können. Dazu besitzen sie einen Adapter, der die jeweils darüber liegende Schicht kapselt. Auf der **physikalischen Schicht** werden elektronische Signale über ein Kabel oder per Funk in einer gewissen Geschwindigkeit in Form von Nullen (keine Spannung) und Einsen (Spannung in Volt) übertragen. Diese sind abhängig vom

<sup>40</sup> Vgl. Tanenbaum, Andrew und Van Steen, Martin (2003): Ebenda, S. 84

<sup>41</sup> Jazaheri, Prof. Medi (2005): „Distributed Systems - Lecture 2: Communication“, Folie 16

<sup>42</sup> Erstens sind diese Protokolle weniger verbreitet und werden zweitens für die Ajax Literaturdatenbank nicht in Betracht gezogen. Sie sollen deshalb hier nur der Vollständigkeit halber erwähnt werden.

<sup>43</sup> Vgl. ISO – International Standardization Organisation und IEC – International Engineering Consortium (1994): „Information technology – Open Systems Interconnection – Basic Reference model: The Basic Model“, S. 29

Kabelmaterial<sup>44</sup> bzw. Art der Funkverbindung<sup>45</sup>, Größe und Form des Netzwerksteckers und der Anzahl bzw. Belegungen der Pins. Die **Sicherungsschicht** überwacht etwaig auftretendes Leitungsrauschen oder andere Fehler mit einer Prüfsummenberechnung, um den Sendevorgang bei Bedarf zu wiederholen. Wenn sich Sender und Empfänger in einem einzelnen Netzwerk, wie z. B. Ethernet, Token Ring oder FDDI, befinden, reichen die beiden untersten Schichten aus. Um heterogene Netzwerke verbinden zu können, wurde die **Vermittlungs- bzw. Netzwerkschicht** eingeführt. Ein Netzwerk wird mit einem Gateway (Übergabestelle) ausgestattet, der eine Gruppe an Hosts und anderen Gateways verwaltet. Diese Übergabestellen bilden, separat betrachtet, wiederum eigene Netzwerke, die die Wegwahl (Routing) einer Nachricht an das Zielnetzwerk koordinieren. Um die Zustellung an einen bestimmten Empfänger zu gewährleisten, wird dieser mit einer eindeutigen globalen Kennung (IP Adresse) versehen. Die Evaluierung einer (schnellen) Route stellt auf dieser Schicht die größte Herausforderung dar, weil sie nicht im Vorhinein geplant wird, sondern spontan von Knoten zu Knoten „hüpft“, wie die nächste Abbildung veranschaulicht:

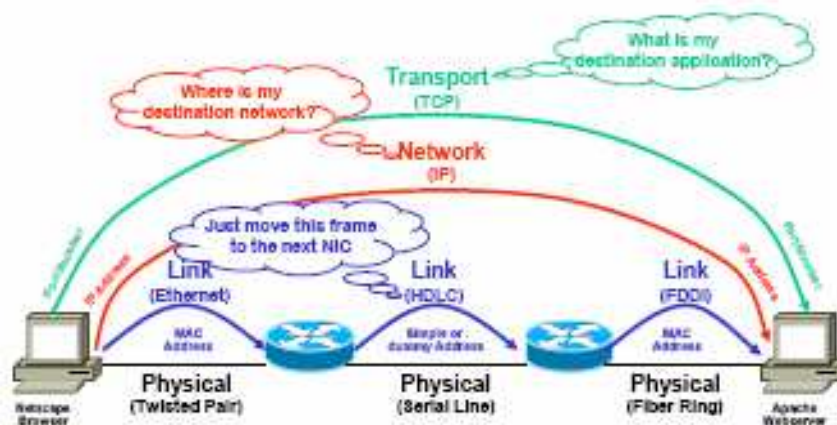


Abbildung 2: Kommunikation über verschiedene Netzwerke<sup>46</sup>

Eine Nachricht wird aufgrund ihrer Größe in mehrere IP Datenpakete zerstückelt, die unterschiedliche Wege nehmen können und daher meist nicht in der richtigen Reihenfolge beim Empfänger ankommen. Das ist eine der Aufgaben der **Transportschicht**, die dafür sorgt, dass die Pakete wieder ordnungsgemäß zusammen gebaut bzw. fehlende Pakete entdeckt werden. Des Weiteren wird mit TCP eine zuverlässige Verbindung geschaffen, die die Nachricht an die verarbeitende Applikation des Empfängers übergibt. Die

<sup>44</sup> Die Übertragung bei Ethernet basiert auf Kupferdrähten (Koaxial), während FDDI Glasfaser benutzt

<sup>45</sup> Mobile Technologien sind u. a. GSM/GPRS, UMTS, WiFi, Bluetooth

<sup>46</sup> Jazaheri, Prof. Medi (2005): Ebenda, Folie 17-24



**Applikationschicht**, welches in diesem Fall das „Hyper Text Transfer Protocol“ (HTTP)<sup>47</sup> ist, führt auf dieser Basis ein Gespräch in Form eines abwechselnden, synchronen Dialogs. Dazu weist HTTP den Teilnehmern die Rollen von **Client** und **Server** zu. Der Client schickt Anfragen, die vom Server beantwortet werden. Die **mentale Schicht** bezeichnet nun die Wahrnehmung des Menschen, d. h. wie die ausgetauschten Nachrichten interpretiert werden. Welche Symbolik sie im World Wide Web (WWW) erhalten und mit welchen Werkzeugen der Client ausgestattet wird, ist der Inhalt des folgenden Abschnitts.

### 2.1.2 Surfen im World Wide Web (WWW)

Die ursprüngliche Idee des WWW war es, wissenschaftliche Dokumente, sei es in Form von Texten, Bildern, Skizzen, Zeichnungen etc., über ein weltweit verfügbares, dezentrales Netzwerk zur Verfügung zu stellen. Damit sollte jedem Forscher der Zugriff auf das gesamte Wissen seines Fachgebiets zugänglich gemacht werden. Dieser Ansatz begeisterte aber nicht nur die Wissenschaft, sondern fand auch in vielen anderen Branchen großen Anklang. Zur Schaffung und Wahrung einheitlicher Standards entstand 1994 die wohl bekannteste Institution des WWW, nämlich das World Wide Web Consortium (W3C), das sich bis heute um die Interoperabilität von Spezifikationen, Richtlinien, Software und Werkzeugen bemüht.<sup>48</sup> Diese Organisation hat zu kontinuierlichen Verbesserungen des WWW geführt und maßgeblich zu dessen großen Erfolg und der starken Verbreitung beigetragen.

Die Möglichkeiten des WWW beruhen auf den grundlegenden Eigenschaften des HTTP Protokolls. Mit dieser Applikation können Dokumente des Servers geholt, gespeichert, gelöscht, Daten hinzugefügt oder deren Meta-Informationen übermittelt werden. Ohne Hilfsmittel müsste der Benutzer noch selbst Befehlsketten kodieren, was jedoch recht unkomfortabel wäre und dem ungeschulten Anwender anfangs recht schwer fallen würde. Deshalb wurde das WWW im Laufe der Zeit mit einigen Werkzeugen ergänzt, die die Client Kommunikation erleichtern. Die Wichtigsten sollen nachstehend vorgestellt werden.

Ein **Uniform Resource Locator (URL)** spezifiziert eine Adresse, unter der ein Dokument gefunden werden kann. Dies ist möglich, weil jedem Rechner im Internet eine eindeutige IP

---

<sup>47</sup> FTP, SMTP etc. werden wie angesprochen nicht berücksichtigt, da sie für das Projekt nicht relevant sind

<sup>48</sup> Vgl. W3C (2007): „World Wide Web Consortium“

Nummer zugewiesen wird, wie z. B. 234.232.222.100. Da diese Adressierung zwar für Maschinen, nicht jedoch für Menschen, leicht lesbar ist, wurde der **Domain Name Service (DNS)** eingeführt. Dieser Service kann mit einer IP-Adresse beliebig viele Namen assoziieren, wie z. B. [www.yahoo.de](http://www.yahoo.de). Der Aufruf eines Dokuments erfolgt über den freigegeben Verzeichnispfad, wie z. B. <http://234.232.222.100/Diplomarbeit.txt> bzw. <http://www.meinRechner.org/Diplomarbeit.txt>. Hingegen ist ein **Uniform Resource Identifier (URI)** eine eindeutige Kennung einer Ressource, wie z. B. [foo://diplomarbeit@gottfried-nindl.net](http://foo://diplomarbeit@gottfried-nindl.net), dessen Ort durch eine URL spezifiziert werden kann. Eine URL ist demnach eine Spezialform eines URIs.<sup>49</sup>

Mit einem **Browser** kann ein Dokument, welches mit einer URL aufgerufen wird, angezeigt werden. Des Weiteren können mit seiner Hilfe Daten aus Formularen an den Server gesendet werden. Die Darstellung erfolgt in Hypertext (HTML), welches neben formatierbaren Text- und Listenelementen auch andere Dokumenttypen, wie z. B. Bilder, Musik, oder Videos, anzeigen kann. Diese werden mit den **MIME-Typen**<sup>50</sup> (Multipurpose Internet Mail Extensions) beschrieben. Die Interpretation anderer Inhalte kann durch **Plugins**, wie z. B. PDF, Shockwave, Flash oder Java, bewerkstelligt werden, die Programme des Betriebssystems einbetten. Darüber hinaus bietet der Browser viele, nützliche Funktionen, wie z. B. Zoomen, Setzen von Lesezeichen, einer Historie oder Druckunterstützung. Die Einfachheit und der Komfort zugleich machen den Browser nicht umsonst zum zentralen Instrument der Benutzersteuerung durch das WWW.

Das wichtige an einem Dokument ist, dass es Verweise zu anderen Dokumenten beinhalten kann, so genannte **Hyperlinks**<sup>51</sup>. Diese speziell gekennzeichneten Elemente können ein anderes Dokument vom Server anfordern. Das Auswählen eines Links löst eine HTTP Methode aus, die an den User Agenten, z. B. einen Browser, delegiert wird. Diese Eigenschaft hat maßgeblich dazu beigetragen, dass Benutzer im Internet „surfen“ können. Es reduziert das manuelle Tippen von URLs erheblich und bringt die Benutzer schneller und einfacher an das gewünschte Ziel.

---

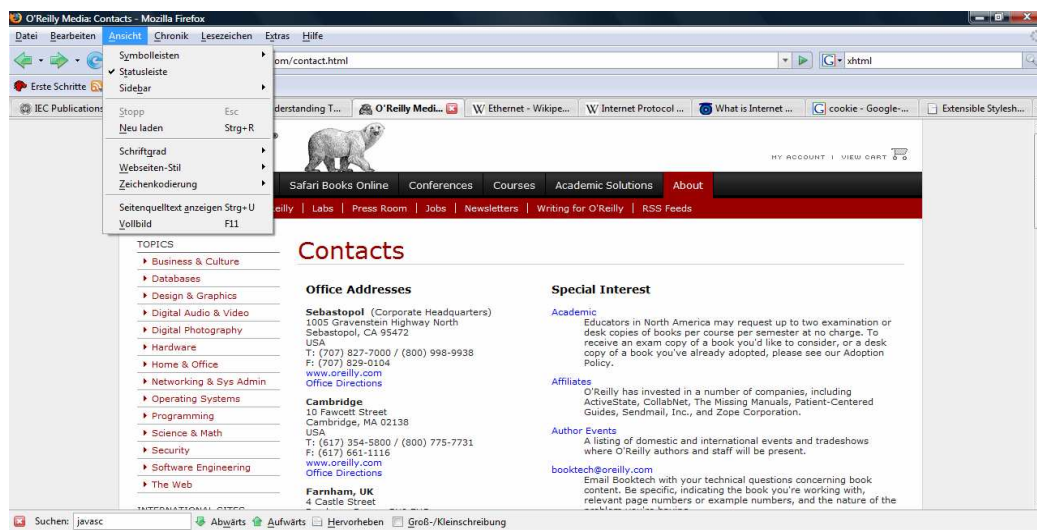
<sup>49</sup> Vgl. W3C (2007): „Naming and Addressing“

<sup>50</sup> Vgl. Swen Wacker (2007): „SELFHTML – MIME-Typen“

<sup>51</sup> Vgl. W3C (2007): „Hyperlinks“

Mit der **Hyper Text Markup Language (HTML)**<sup>52</sup> können Dokumente mit Hilfe von Schlüsselwörtern in verschiedene Unterabschnitte gegliedert werden. Dadurch können mehrere Dokumenttypen (MIME-Typen) in einem Dokument angezeigt werden. Darüber hinaus bietet HTML Darstellungselemente, wie z. B. Bäume, Tabellen, Formulare, Hyperlinks etc., die zur Strukturierung der Inhalte verwendet werden können. **Cascading Style Sheets (CSS)**<sup>53</sup> können die Elemente mit Schriftarten, Farben etc. formatieren. **Cookies** sind persistente Daten des Benutzers, die für automatische Identifizierung und Sitzungen eingesetzt werden. **JavaScript**<sup>54</sup> ist eine Add-On Programmiersprache des Browsers, die erweiterte Funktionalitäten zur Unterstützung der Benutzerführung bereit stellt. Damit kann auf der Clientseite z. B. dynamisches HTML bzw. Grafiken erzeugt oder das Sicherheitsprofil von Cookies abgefragt werden. Zur besseren Trennung von Präsentation und Daten und zur Beschreibung von Metadaten werden noch **XHTML**<sup>55</sup>, **XML**<sup>56</sup> oder **XSLT**<sup>57</sup> eingesetzt.

Um eine aktuelle HTML Webseite in einem modernen Browser anzuzeigen, werden die genannten Technologien in Kombination eingesetzt (siehe Abbildung 3). Dies stellt aber nur die oberflächlichste Ebene des Client dar. Der Server hingegen arbeitet im Hintergrund, um einerseits Funktionalität, wie z. B. der Zugriff auf eine Datenbank, und andererseits eine geeignete Darstellung für den Client, wie z. B. bei einem Browser in HTML, zu unterstützen. Dieses Zusammenspiel wird im folgenden Abschnitt näher ausgeführt.



**Abbildung 3: Browserfenster mit HTML Seite**

<sup>52</sup> Vgl. W3C (2007): „HTML 4.01 Specification“

<sup>53</sup> Vgl. W3C (2007): „Cascading Style Sheets“

<sup>54</sup> Vgl. Wacker, Swen (2007): „JavaScript“

<sup>55</sup> Vgl. W3C (2007): „XHTML 1.1“

<sup>56</sup> Vgl. W3C (2007): „Extensible Markup Language“

<sup>57</sup> Vgl. W3C (2007): „Extensible Stylesheet Language“

## **2.2 Webapplikationen und -services**

Die Tatsache „Originally developed as a document-sharing platform, the web is still often considered as such“<sup>58</sup> bestätigt einerseits die erfolgreich umgesetzte Idee, bedeutet andererseits eine Beschränkung des WWW auf eine simple Datenbank. Das WWW wurde aber im Laufe der Zeit zu weit höheren Aufgaben berufen. Kaufhäuser, Auktionen, Diskussionsrunden, Email-Dienste, Radiostationen, Zeitungen u. v. m. halten seit den späten 90er Jahren Einzug in die virtuelle, globale Welt. Dies erfordert mehr als einfache, lesende Applikationen, die statische Webseiten als Museumsstücke ausstellen.<sup>59</sup> Das Realisieren einer dauerhaften Sitzung, die viele, personalisierte Anfragen bzw. Antworten über einen längeren Zeitraum verarbeiten können, übersteigt die Fähigkeiten des statuslosen HTTP Protokoll. Hinzu kommt das die Dokumentsprache HTML nur passive Schnittstellen zu externen Datenquellen besitzt und keine aktive Laufzeitkomponente, die auf Anfrage dynamisch Information, wie z. B. aus Datenbanken, erzeugen kann. Da im Internet von heute noch kein einheitliches Protokoll für die Sitzungs- und Darstellungsverwaltung existiert, wird dieser Prozess auf der Client- und Serverseite mit diversen Hilfsmitteln nachsimuliert. Diese sind besser bekannt unter dem Sammelbegriff „Webtechnologien“ und unterstützen die Kommunikation mit mächtigen Werkzeugen, die in der Lage sind, interaktive Applikationen zu gestalten. Welche Entwurfsmuster sich daraus ergeben können, sollen die folgenden Unterkapitel beschreiben.

### **2.2.1 Client und Server**

Webtechnologien ergänzen HTTP und HTML dort, wo ihre Grenzen liegen. Ein Server, der Information bzw. Dienste anbietet, hat gegenüber einem Client, der diese nutzt, andere Aufgaben bzw. Probleme zu bewältigen. Während auf der Serverseite die Datengenerierung und die Bereitstellung von Applikationsfunktionen im Vordergrund stehen, beschäftigt sich die Clientseite mit der Informationsdarstellung und Benutzerführung. Auf Basis dieser Rollenverteilung wurden im Laufe der Zeit eine Reihe von Technologien entwickelt, die eine 2 Schichten (2-tier) Architektur von Webapplikationen unterstützen:

---

<sup>58</sup> Hassan, Ahmed E. und Holt, Richard C. (2002): „Architecture Recovery of Web Applications“, S. 349

<sup>59</sup> Vgl. Kappel, Gerti et al (2001): „Modeling Customizable Web Applications“, S. 168

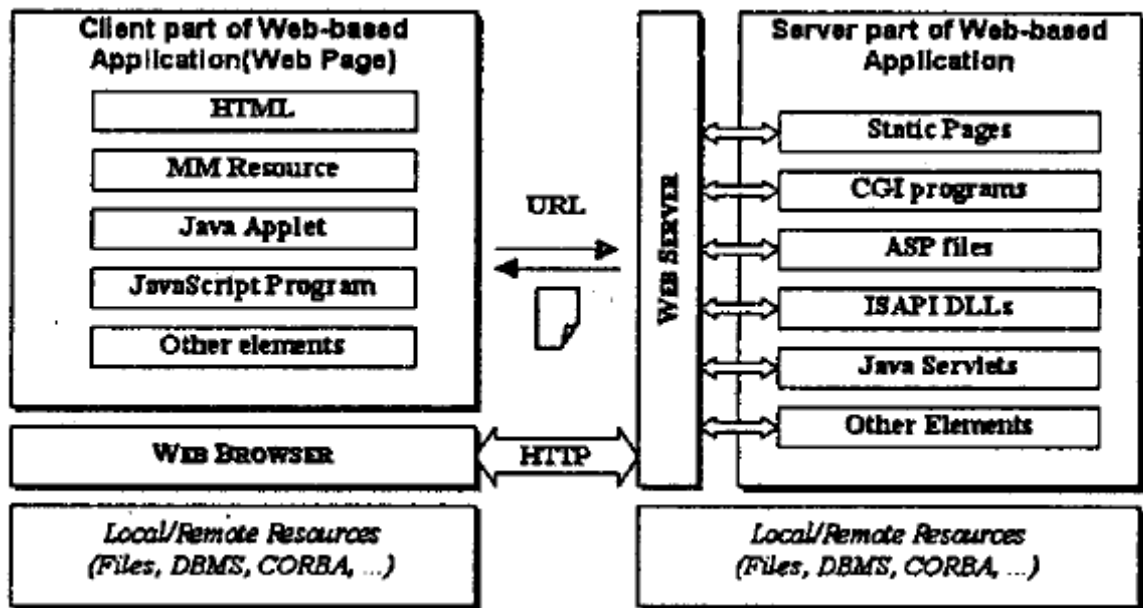


Abbildung 4: Architektur webbasierter Applikationen<sup>60</sup>

Die einzelnen Server und Client Komponenten funktionieren unabhängig voneinander und können fast beliebig kombiniert werden. Zum Beispiel können auf der Serverseite ein ASP/PHP Script und gleichzeitig auch ein CGI Programm zur Generierung von HTML laufen. Auf der Clientseite kann ein Java Applet, das von einer JavaScript Methode eingebunden wird, von einem Java Servlet oder auch von einer statischen Seite aus aufgerufen werden. Dies erlaubt ein flexibles Austauschen, Adaptieren, Hinzufügen und sogar Kapseln von Komponenten. Damit versuchen Webtechnologien zwei offensichtliche Schwächen des WWW zu kompensieren: Einschränkungen in der Benutzeroberfläche und im Datenzugriff. Während sich die Clientseite mit Präsentationslogiken, wie z. B. HTML, Java Applets oder JavaScript Programmen, befasst, übernimmt die Serverseite die Aufgabe der Geschäftslogik, wie z. B. dynamische Abfragen aus Datenbanken. Dadurch können Anwendungen im Web, wie z. B. Email-Programme oder Online-Shops, konstruiert werden, die applikationsspezifische Funktionen zu einem Produkt bündeln. Auf Softwareebene kann das Client-Server Modell (2-tier) Erweiterungen auf 3-tier und n-tier Architekturen erfahren. Als nächstes wird jedoch ein Ansatz vorgestellt, in dem jeder Teilnehmer Server und Client zugleich ist.

<sup>60</sup> Zhao, Weiquan und Chen, Jian (1999): „A Component Oriented Web Application Model“, S. 192

## 2.2.2 Peer-to-Peer (P2P)

Im Gegensatz zum Client-Server Modell sind in einem Peer-to-Peer Netzwerk alle Teilnehmer gleichberechtigt. Jeder Knoten in einem Netzwerk besitzt eine bidirektionale Verbindung zu anderen Knoten, und fungiert als Server und Client zugleich. Die wichtigsten Eigenschaften von Peer-to-Peer definieren sich wie folgt:<sup>61</sup>

- *Rollensymmetrie*: Jeder Peer ist Client und Server.
- *Dezentralisierung*: Es existiert keine statische, globale Sicht.
- *Selbstorganisation*: Das Verhalten der einzelnen Peers steuert sich selbst und muss nicht von einer Instanz kontrolliert werden.
- *Autonomie*: Peers sind unabhängig voneinander und agieren auch so.
- *Verfügbarkeit*: Das System ist fehlertolerant und kann Ausfälle von Verbindungen kompensieren.

Es können drei Typen von Peer-to-Peer (P2P) Systemen unterschieden werden: zentrale, dezentrale und hybride. In den Anfängen der P2P Systeme waren noch zentrale Systeme, wie z. B. Napster, verbreitet. Im Laufe der Zeit liefen ihnen aber dezentrale, wie z. B. Freenet, oder hybride Systeme, wie z. B. Gnutella, aufgrund der besseren Skalierbarkeit den Rang ab. Die Architektur von P2P Systemen ist recht unterschiedlich, bewegt sich aber im Wesentlichen auf vier Ebenen:<sup>62</sup>

- *Netzwerk*: Transport über das Netzwerk, z. B. mit TCP/IP
- *Datenzugriff*: Auffinden von Ressourcen, z. B. mit dem Gnutella Netzwerk
- *Dienstebene*: Erweiterte Dienstleistungen, wie z. B. Telefonieren mit Skype
- *Benutzerebene*: Communities, die das System benutzen, wie z. B. Ebay

Ein Architekturstandard ist zurzeit noch nicht vorhanden, welcher aber mit dem Open Source Projekt JXTA<sup>63</sup> verfolgt wird. JXTA bedient sich offener Protokolle, um Geräte über ein Netzwerk zu verbinden. Das gleiche Prinzip gilt auch für Webservices, die als Alternative bzw. Ergänzung nahtlos an unterschiedlichen Stellen einer Infrastruktur eingebaut werden können. Das Prinzip von Webservices wird im nächsten Abschnitt erläutert.

---

<sup>61</sup> Vgl. Hauswirth, Manfred und Dustdar, Schahram (2007): „Peer-to-Peer: Grundlagen und Architektur“, S. 2

<sup>62</sup> Vgl. Hauswirth, Manfred und Dustdar, Schahram (2007): Ebenda, S. 2f

<sup>63</sup> Vgl. JXTA(TM) Community Project (2007): „JXTA“

### 2.2.3 Webservice

Ein Webservice ist ein Dienst, der im WWW mit einem URI eindeutig identifiziert wird. Dazu existiert eine Schnittstellenbeschreibung in der „Web Service Description Language“ (WSDL), die die Methoden und deren Parameter in Form der XML Syntax spezifiziert. In einem Verzeichnisdienst „Universal Description, Discovery and Integration“ (UDDI) können mit dieser Metainformation Webservices registriert und gesucht werden. Die eigentliche Übermittlung verwendet das „Simple Object Access Protocol“ (SOAP), welches den Nachrichtenaustausch in Form von Objekten ermöglicht.<sup>64</sup>

Ein Webservice dient als Schnittstelle zwischen Maschinen. Dadurch soll es Applikationen möglich gemacht werden, Dienste zur Laufzeit zu veröffentlichen, zu suchen und auszuführen. In der ursprünglichen Auffassung werden Webservices als Verbrauchsprodukt betrachtet, d. h. dass der Service Benutzer eine Anfrage an den Service Provider schickt, der dann mit einer entsprechenden SOAP Nachricht antwortet.<sup>65</sup> Eine neuere Entwicklung ist das Konzept eines Zwei-Weg-Webservice, in der ein Webservice-Endpunkt als Client und Server zugleich fungiert, ähnlich wie es bei einem P2P Netzwerk der Fall ist. Die Kommunikation zwischen den eigentlichen Teilnehmern kann dadurch asynchron gestaltet werden, wie die nächste Abbildung u. a. zeigt:



Abbildung 5: Zwei-Weg-Webservice<sup>66</sup>

Ein Zwei-Weg-Webservice führt eine zusätzliche Instanz zwischen Client und Server ein. Ein modifizierter Webservice dieser Art wird für folgende Aufgaben vorgesehen:<sup>67</sup>

<sup>64</sup> W3C (2000): „Simple Object Access Protocol (SOAP) 1.1“

<sup>65</sup> Vgl. Chou, Wu et al (2006): „Web Services for Service-Oriented Communication“, S. 3

<sup>66</sup> Vgl. Lui, Feng et al. (2006): „Web Service for Distributed Communication Systems“, S. 1033

<sup>67</sup> Vgl. Lui Feng et al. (2006): Ebenda, S 1031ff

- Status- bzw. sitzungsorientierte Transaktionen
- Ereignisorientierte, asynchrone Programmierung
- Webservice Router (siehe Abbildung 5)

Webservices zeichnet Flexibilität in der Architektur und eine leichte Portabilität in andere Umgebungen aus. Sie können auf der Serverseite und dank des Zwei-Weg Prinzips auch auf der Clientseite angeboten werden. Zusätzliche Funktionalitäten auf der Clientseite sind vor allem für das Design von „Rich Internet Applikationen“ (RIA) entscheidend, die im Folgenden näher ausgeführt werden.

## 2.2.4 Rich Internet Applikationen (RIA)

Aus Benutzersicht simulieren Webapplikationen eine dauerhafte Sitzungs- und Darstellungsumgebung im Webbrowser. Betriebssystembasierte Anwendungen bieten jedoch im Vergleich zu Webapplikationen einen weit höheren Bedienungskomfort. Das Ziel von RIAs ist daher die Schaffung von benutzerfreundlichen Browserapplikationen, deren Aussehen und Bedienungsweise sich nicht mehr vom Desktop unterscheidet, wie Fartell treffend ausdrückt: „[...] since many people ‘live’ on the desktop, proper Internet applications should be an extension of where they live; of their comfort zone.“<sup>68</sup> Die Motivation entstand aus den Unzulänglichkeiten herkömmlicher Webapplikationen, von denen die wichtigsten nachstehend genannt werden:<sup>69</sup>

- *Prozesse*: Eine einzige Aufgabe erfordert das Navigieren durch eine Vielzahl an Seiten, z. B. um einen Flug zu buchen.
- *Daten*: Suchen nach Daten ist meist mühsam und bringt aufgrund schlechter Trennung von Präsentation und Daten nur mäßige Resultate.
- *Konfiguration*: Das Verändern der Einstellungen ist zwar für Entwickler nachvollziehbar, jedoch ist es nicht unbedingt für die Anwender intuitiv.
- *Feedback*: Ohne erneute Anfrage an den Server kann der Benutzer keine Rückmeldungen erhalten. Die Interaktivität ist dadurch sehr eingeschränkt.

---

<sup>68</sup> Fartell, Jason und Nezek, George S. (2007): „Rich Internet Application – The Next Stage of Application Development“, S. 413

<sup>69</sup> Vgl. Preciado, J. C.(2005): „Necessity of methodologies to model Rich Internet Applications“, S. 2



RIAs benötigen clientseitige Komponenten, wie z. B. eine GUI Bibliothek, ein Event Handling, einen Layout Manager oder ein Datenmodell, die den Browser mit zusätzlichen Funktionsbibliotheken ausstatten. Es werden dabei drei unterschiedliche Ansätze verfolgt:<sup>70</sup>

- *Browser-Plugin:* Mittels der Installation einer betriebssystemspezifischen Laufzeitumgebung können auf einem Browser externe Programme ausgeführt werden. Der erste Durchbruch war das Java-Applet, dem Flash, Shockwave, PDF, Adobe Flex 2, Java Webstart oder Microsoft Smart Client folgten. Der gesamte Applikationscode wird bei der ersten Anfrage vom Server an den Client übertragen und ausgeführt.
- *Ajax:* Die Technologien XHTML, HTML, XML, DHTML, CSS, DOM, Cookies und JavaScript werden kombiniert, um eine interaktive Benutzeroberfläche auf dem Browser zu bauen. Eine Ajax-Engine auf der Clientseite sorgt dafür, dass nur mehr benötigte Daten, und nicht mehr ganze Dokumente, bei Bedarf nachgeladen werden.
- *Browser-basiert:* Hierbei handelt es sich um eine Benutzerschnittstellensprache, die die benötigten Elemente, sowie deren Interaktionen, deklariert. Ein Beispiel dafür ist die „XML User Interface Language“ (XUL) von Mozilla, die auf existierenden Standards, wie XML, CSS, DOM oder JavaScript aufbaut. Ein Mozilla Browser, wie z. B. Firefox, kann XUL richtig interpretieren und darstellen.

Bei Ajax handelt es sich um die jüngste Technologie, der eine verheißungsvolle Zukunft prophezeit wird.<sup>71</sup> Sie soll sogar das Potential besitzen, den bestehenden Desktop gänzlich zu ersetzen.<sup>72</sup> Zur Überprüfung dieser Behauptung wird im Zuge dieser Arbeit die Ajax Technologie mit dem BOC Forschungsprojekt ADOweb kombiniert, um eine browserbasierte RIA zu realisieren. Das Projekt ADOweb wird im nächsten Abschnitt thematisiert.

## **2.3 ADOweb**

Der rapide Aufschwung des WWW in den 90er Jahren weckte das Interesse traditionell gewachsener Softwarehäuser, die neuen Distributionswegen der weltweiten Vernetzung zu nutzen. Diese Entwicklung erkannte auch die BOC, welche eine 1995 gegründete Spin-Off

---

<sup>70</sup> Fartell, Jason und Nezek, George S.: Ebenda, S. 415

<sup>71</sup> Vgl. Powell, Thomas (2006): „AJAX is the future of Web app development“

<sup>72</sup> Vgl. Charland, Andre (2007): „Will AJAX Replace the Desktop?“

Firma des DKE Instituts der Universität Wien ist.<sup>73</sup> Die BOC ist ein Beratungs- und Softwareunternehmen, welches sich auf Management Lösungen mit IT Unterstützung spezialisiert hat.<sup>74</sup> Es werden Modellierungsumgebungen in Form von

- *ADONIS und PROfit* (Geschäftsprozessmanagement),
- *ADOit* (IT Architektur- und Servicemanagement),
- *ADOScore* (Strategische Planung mit Balanced Scorecards),
- und *ADOLOG* (Supply Chain Management)

für die Betriebssysteme Windows, OS 2 und Linux entwickelt bzw. vertrieben. Das Projekt ADOWeb wurde nun aus dem Grund ins Leben gerufen, um die Stand-Alone Produkte der BOC über das Web verfügbar zu machen. Grundsätzlich werden dabei drei Lösungsansätze in Betracht gezogen:<sup>75</sup>

- Exportieren der Information in ein für den Client verständliches Format, wie z. B. HTML
- Remote Desktops, wie z. B. Citrix Metaframe, die den Bildschirm übertragen
- Installation eines eigenen Servers

Die zweite Variante wird ausgeschlossen, da es entweder das Erwerben von Lizenzen erfordert oder auf Marktlösungen mit unzureichender Funktionalität zurückgegriffen werden muss. Da die dritte Möglichkeit nur mit viel Aufwand skalierbar ist, wird in der firmeninternen Forschung der erste Vorschlag verfolgt. Der aktuelle Stand der Entwicklung (State-of-the-art) und die Einordnung der Ajax Literaturdatenbank werden in den folgenden Unterkapiteln erläutert.

### **2.3.1 Positionierung**

Die ursprüngliche Idee von ADOWeb war der Export von Modellinformationen aus ADO-Produkten. Darum ist es traditionell auf höheren, dem Client näheren, Ebenen angesiedelt. Die nächste Abbildung zeigt die Positionierung von ADOWeb im Kontext der Produktfamilie:

---

<sup>73</sup> Vgl. Hawlik, Wolfgang (2001): „Symbiose zwischen Forschung & Anwendung“, S. 21

<sup>74</sup> Vgl. BOC (2007): „Company profile of the BOC group“

<sup>75</sup> Vgl. Woitsch, Robert (2005): „ADOWeb – Overall Architecture and Definition 1.0B“, S. 5

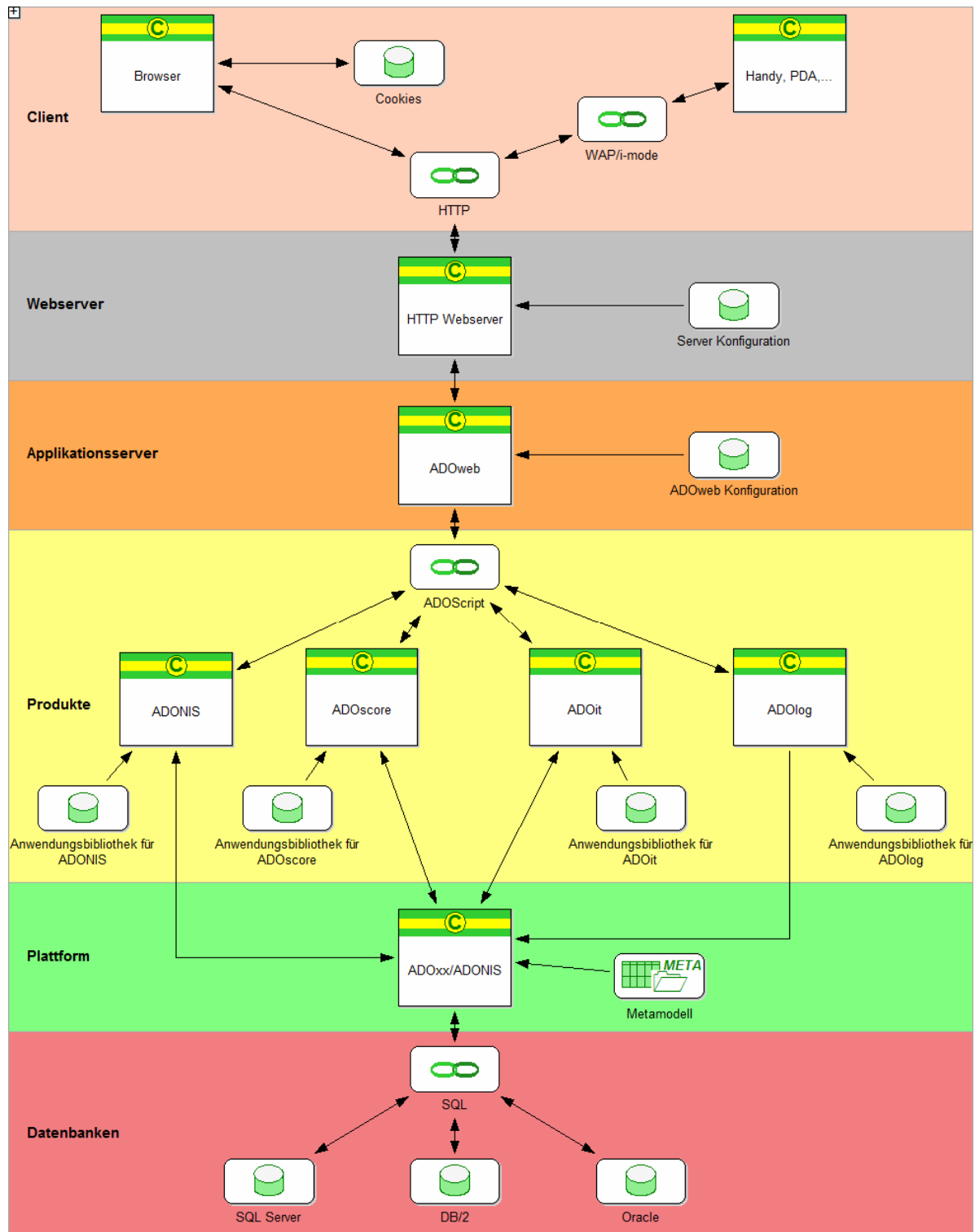


Abbildung 6: Positionierung von ADOweb in die ADO-Produktfamilie

Auf der untersten Ebene befinden sich die **Datenbanken**. Alle möglichen Ressourcen werden dort dauerhaft abgespeichert, wie u. a. Modelle und deren Objekte bzw. interne Skripten. Der Zugriff auf die Datenbank geschieht über die **Plattform**, die mittels der Abfragesprache SQL („Standard Query Language“) Datenbestände abrufen, aktualisieren oder löschen kann. Die

Plattform befindet sich zurzeit im Umbau, da ADOxx langfristig die bestehende Version ADONIS ablösen wird. Abgesehen davon besitzen beide Versionen ein Metamodell, welches abstrakte Entitäten, wie z. B. Start, Aktivität oder Ressource etc., enthält, die die Grundelemente für die Konstruktion eines Modells bereit stellen. Die **Produkte** bestehen aus einer Anwendungsbibliothek, die spezifische Modelle für eine bestimmte Domäne, wie z. B. IT-Architekturen, umfassen. Mittels „Customizing“<sup>76</sup> können die Modell- und Objekttypen angepasst bzw. beliebig kombiniert werden, um einen hohen Grad an Personalisierung zu erreichen. Der **Applikationsserver** greift mit Hilfe von Webservices, die intern mit der Makroskriptsprache AdoScript kommunizieren, auf die Modellbestände zu. AdoScript bietet darüber hinaus ein mächtiges Repertoire an Funktionen, wie z. B. dem Realisieren neuer Schnittstellen.<sup>77</sup> ADOweb ist nun dafür verantwortlich, dass die Information in einem webverständlichen Format aufbereitet wird, damit sie vom **Webserver** verteilt werden kann. Dieser kommuniziert über eine HTTP Verbindung (siehe Abschnitt 2.1.1) mit den **Clients**, die vorrangig Webbrowser sind. In Zukunft sollen aber auch Geräte unterstützt werden, die über eine WAP bzw. i-mode Schnittstelle verfügen.

ADOweb ist somit in erster Linie ein Publikationswerkzeug für ADONIS. Dies ist aber bei Weitem nicht alles. ADOweb hat auch das Potential, Daten dynamisch aus dem Web zu beziehen und somit als Konfigurator der Plattform zu fungieren. Nach Außen hin ist eine große Anzahl an Distributionskanälen, sei es über Webbrowser, Mobiltelefonen, PDAs etc. denkbar. Um diese Generierung zu gewährleisten, bedarf ADOweb eines mehrschichtigen Aufbaus, der im folgenden Abschnitt vorgestellt wird.

### 2.3.2 Architektur

Die Architektur von Informationssystemen lässt sich auf drei konzeptionellen Ebenen beschreiben: *Präsentation, Geschäftslogik und Ressourcenmanagement*.<sup>78</sup> Ähnlich zu dem in Kapitel 2.1.1 vorgestellten Kommunikationsmodell (Abbildung 1) werden gewisse Aspekte der Verarbeitung einzelnen Schichten zugeordnet, damit die Komplexität eines Systems in einfachere Teile gefasst wird („Teile und Herrsche“ Prinzip). ADOweb orientiert sich an

---

<sup>76</sup> Abbildung 6 zeigt die Benutzersicht, in der die Anwendungsbibliotheken nicht verändert werden können. Als Administrator erhält man mit dem Administrations Toolkit Kontrolle darüber.

<sup>77</sup> Vgl. Scheper, Ulrich Mag. (2007): „ADONIS – Customizing“, Kap. AdoScripts & Expressions

<sup>78</sup> Vgl. Alonso, Gustavo et al (2004): „Web Services“, S. 4

dieser 3-Ebenen Abstraktion. Die folgende Grafik gibt einen groben Überblick über die Architektur des ADOweb Service Frameworks:

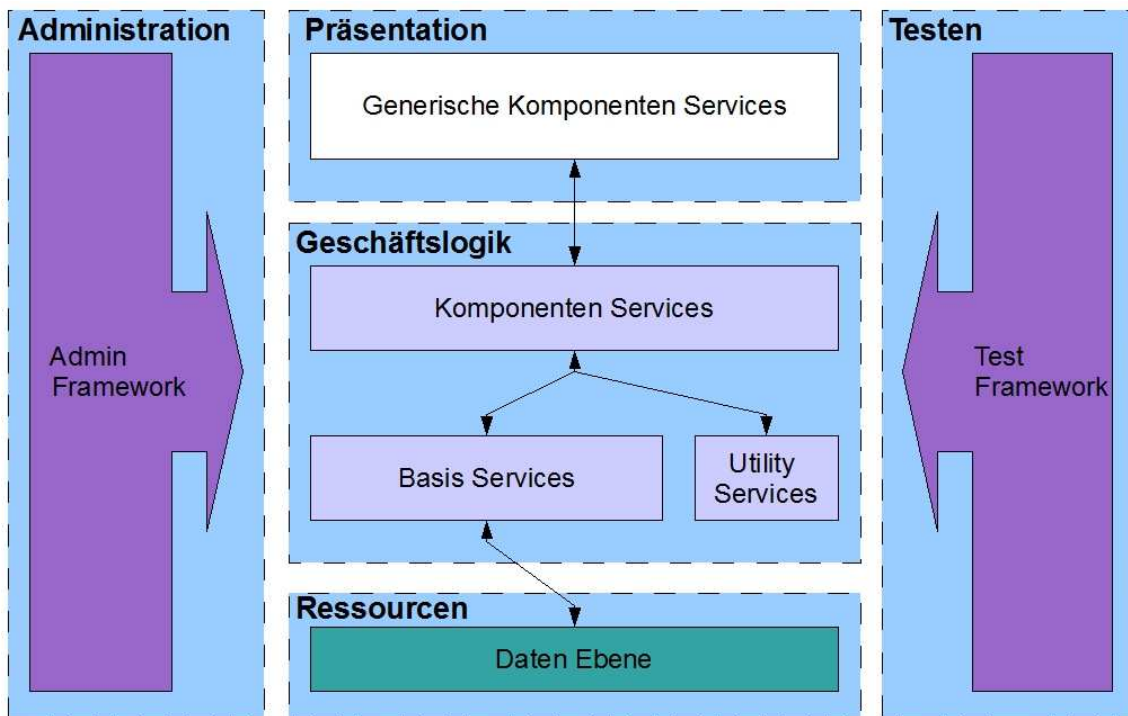


Abbildung 7: ADOweb Framework<sup>79</sup>

Die generischen Komponenten Services sind dafür verantwortlich, dass eine Benutzeroberfläche (**Präsentation**) in einem stark verbreiteten Webbrowser, wie MS Internet Explorer oder Mozilla Firefox, geschaffen wird. Dazu werden je nach Bedarf die in Abschnitt 2.1.2 erwähnten Webtechnologien, wie u. a. HTML, JavaScript, XML, XSLT oder Java Applets, verwendet. Der eigentliche Inhalt der generischen Komponenten Services befindet sich in den Komponenten Services, die die Funktionalität eines bestimmten Anwendungsszenarios unterstützen (**Geschäftslogik**). Die wichtigsten Komponenten werden exemplarisch in der folgenden Tabelle kurz vorgestellt:<sup>80</sup>

Komponente	Beschreibung	Technologie
<i>Web Model Editor</i>	Ein Editor zum Manipulieren von Modellen	Java-Applet
<i>Web Model Viewer</i>	Darstellung und Information von Modellen	HTML, XML, XSLT, Struts

<sup>79</sup> Vgl. Eichner, Hannes Dipl.-Ing. (2007): „ADOweb Architektur“

<sup>80</sup> Vgl. Woitsch, Robert (2005): „ADOweb – Overall Architecture and Definition 1.0B“. S. 26f

<i>Web Import/Export</i>	Ein- und Auslesen von Modellen in ADL, XML oder XMI Format	HTML, XML, XSLT, Struts
<i>Web Workbench</i>	Arbeitsoberfläche mit Login, Navigation und Integration von anderen Komponenten	HTML, JavaScript, Spring XML
<i>Web Content Management</i>	Veröffentlichung und Verwaltung von Information	HTML, Cocoon

**Tabelle 2: Wichtige ADOweb Komponenten**

Die Komponenten Services kapseln die Geschäftslogik einer Anwendung. Sie sind benutzeroberflächenunabhängig und stellen Adapter bereit, auf die eine Benutzeroberfläche aufgesetzt werden kann. Verarbeitungsroutinen, wie z. B. Datenaggregation, Caching, Logging oder Sicherheitsmechanismen, werden von den Basis Services verrichtet, auf die die Komponenten Services zurückgreifen können. Die Utility Services hingegen werden für Aufgaben, die unabhängig vom Anwendungsszenario jedem Service zur Verfügung stehen sollen, wie z. B. dem Bibliotheksmanagement, benötigt. Die unterste Ebene ist für die Verwaltung der Daten (**Ressourcen**) zuständig. Diese Aufgabe übernimmt ADONIS, welches einen Webservice (AdoScriptService) anbietet, der AdoScript Kommandos ausführen kann. Zusätzlich können auch andere Datenbanken bzw. Server, wie z. B. ein Repository Server, eingesetzt werden. Zur Erleichterung der Serviceentwicklung und -verwaltung wird das Admin Framework (**Administration**) bereit gestellt. Performanz-, Stabilitäts- und Funktionalitätstests werden vom Testframework (**Testen**) bewerkstelligt. Die Ajax Literaturdatenbank soll sich als Komponente bzw. als generische Komponente wieder finden, die Basis und Utility Services in Anspruch nimmt. Welche Absichten hinter diesem Projekt liegen, führt der nächste Abschnitt näher aus.

### 2.3.3 Web 2.0 mit PROMOTE

Die aktuelle Version von ADOweb 0.8 schöpft aus den traditionellen Möglichkeiten des WWW. Unter der Verkleidung des Projekts einer wissensbasierten Literaturdatenbank steckt die Ambition eine PROMOTE Applikation zu entwerfen, die dem im Kapitel 2.4 vorgestellten Begriff des Web 2.0 entspricht. Es wird eine für die BOC kaum erprobte Technologie Ajax eingesetzt. Dabei wird das Prinzip der *Leichtgewichtigkeit* verfolgt. Dies

hat zwei einschneidende Auswirkungen auf der Entwicklungs- und Benutzerebene. Zum einen soll die Applikation auf einem gängigen Webbrowser laufen, ohne zusätzliche Installationen bzw. Plugins zu benötigen. Zum anderen sollen die Benutzer ihre eigenen Vorstellungen einbringen können, wie z. B. bei Ebay oder Youtube, die zur Laufzeit eigene, angepasste Modelle generieren. Dabei sind folgende Grundfunktionalitäten von ADOweb relevant, die bereits für die Wissensmanagementmethode PROMOTE entwickelt wurden:

- *Excel Akquisition:* Der Inhalt einer Excel Tabelle wird als Modell oder als eine Liste von Objekten in den Modellbestand importiert. Dadurch kann Excel als Designer für Dokumentvorlagen verwendet werden.
- *Modell Tagging:* Mit einem Modell bzw. Objekt können verschiedenste Text-, Bild- oder andere Mediendateien, Hyperlinks oder Webservices verknüpft werden. Dies soll der intuitiveren Veranschaulichung der abstrakten Modellierung dienen.
- *Webmodeller Light:* Die Eingabe der Modelldaten erfolgt über ein Webformular, wobei auf die grafische Darstellung des Modells auf der Benutzerseite verzichtet wird. Die visuelle Positionierung der Objekte erfolgt mit einem automatischen Algorithmus.
- *Bildhafte Modellierung:* Bilder bzw. Fotos werden von Benutzern ins System gespielt. Diese werden in Bezug auf signifikante Merkmale analysiert bzw. katalogisiert. Damit kann z. B. ein Foto eines Teppichs modellhaft beschrieben werden.
- *Prozess Video:* Die Benutzer stellen ähnlich zu „Bildhafte Modellierung“ Videos in das System. Daraus wird eine schematische Abbildung eines Prozesses erstellt. Somit wird z. B. ein Verkaufsvideo mit einem Wissensmanagementprozessmodell beschrieben.

Die Anwendung der Literaturdatenbank greift auf die genannten Funktionen des Webmodeller Light und der Excel Akquisition zurück. Sie soll von einer Community verwendet werden, die jedes Mitglied dazu berechtigt, Zeitschriftenartikel oder andere interessante Dokumente einzustellen. Zu jedem eingetragenen Dokument kann ein Blog eröffnet werden, um die Meinungen und Bewertungen der Leser aufzunehmen. Das Prinzip eines Blogs ist einer von mehreren Web 2.0 Ansätzen, die im folgenden Abschnitt vorgestellt werden.

## 2.4 Web 2.0

Die erfolgreiche Verbreitung und beliebte Nutzung von Kollaborationsportalen, wie z. B. der Online-Enzyklopädie Wikipedia oder dem Videoportal Youtube<sup>81</sup>, weckten die Aufmerksamkeit namenhafter Fachleute. Der Verlag O'Reilly und der Konferenz- bzw. Messeveranstalter MediaLive International versuchten in einem Brainstorming, die Ursachen dieser gesellschaftlichen Massenphänomene zu analysieren. Die Ergebnisse wurden im Artikel „What is Web 2.0“ veröffentlicht, welcher eine neue, zweite Ausgabe des WWW beschreibt. Eine Versionierung hat in der Softwareindustrie laut Balzert<sup>82</sup> den Zweck, die Ausprägung eines Software Elements zu einem bestimmten Zeitpunkt zu kennzeichnen. Eine Versionsnummer besteht aus zwei Teilen, wobei im Falle von einschneidenden Änderungen im Software-Element die vordere Release-Nummer um 1 erhöht und die hintere Level-Nummer auf 0 gesetzt. Web 2.0 verspricht folglich einerseits eine erfolgreiche Fortsetzung des WWW, die andererseits deren Probleme und Einschränkungen hinter sich lässt. Der erwähnte Artikel „What is Web 2.0“ arbeitet u. a. folgende konzeptionelle Punkte heraus, die eine Abgrenzung zum herkömmlichen Web 1.0 zeigen sollen:<sup>83</sup>

Web 1.0	Web 2.0
Persönliche Webseiten	Blogging
Spekulation über Domainnamen	Optimierung von Suchmaschinen
Seitenaufrufe	Interaktive Benutzerführung
Extraktion mittels Screen Scraping	Webservices
Veröffentlichung	Teilnahme
Verzeichnisse (Taxonomie)	Tagging (Volksmund)
Dominanz einzelner Firmen (Microsoft)	Freie Communitybildung (Eclipse)
Feststehend (Stickiness)	Zusammenwachsend (Syndication)

**Tabelle 3: Konzeptioneller Unterschied Web 1.0 zu Web 2.0**

Diese Liste kann endlos fortgesetzt werden. Zur Wahrung des Überblicks hat Patrick Holz die wichtigsten Konzepte und Prinzipien zusammengefasst:<sup>84</sup>

<sup>81</sup> Vgl. YouTube, LLC (2008): „Broadcast Yourself“

<sup>82</sup> Vgl. Balzert (1998): „Lehrbuch der Software-Technik“

<sup>83</sup> Vgl. O'Reilly, Tim (2005): „What is Web 2.0“

<sup>84</sup> Vgl. Holz, Patrick (2006): „Nutzung kollektiver Intelligenz“



- *Web als Plattform:* Vom GUI zum Entwicklungsstandard
- *Kollektive Intelligenz:* Communities als Expertensysteme
- *Daten als Intel Inside:* Das Web als Datenbank
- *Ende des Softwareentwicklungszyklus:* Benutzer als Mitentwickler von Beta Versionen
- *Leichtgewichtes Programmiermodel:* Technologie und Inhalte lassen sich in lose gekoppelten Systeme beliebig kombinieren
- *Software über Gerätegrenzen hinweg:* iPod und iTunes
- *Rich User Experience:* Ajax, Flash, XUL, Voip, Google Maps, Address Book etc.

Kollektive Intelligenz kann allgemein als das Lösen von Problemen mit Hilfe von dezentralen Informationen bzw. Wissen von verschiedenen Personen und Organisationen definiert werden.<sup>85</sup> Eine Community, die sich selbst organisiert, kommuniziert und kooperativ zusammen arbeitet, ist entscheidend für das Zustandekommen dieses Phänomens. Eine erfolgreiche Web 2.0 Anwendung, wie z. B. das Studentenverzeichnis StudiVZ<sup>86</sup>, basiert auf folgenden drei Grundelementen:<sup>87</sup>

- *Services:* Dienste zur Interaktion mit der Webportal
- *Content:* Inhalte, die zur Verfügung gestellt werden
- *Community:* Beziehungen und Assoziationen der teilnehmenden Personen

Diese Prinzipien sollen am Beispiel des PROMOTE Szenarios der Literaturdatenbank erläutert werden. Die Literaturdatenbank stellt in erster Linie Dienste zum Einstellen, Editieren, Kategorisieren und Suchen von Literatur zur Verfügung. Der Zugang erfolgt ausschließlich über den Browser, wobei die Web-Plattform die gesamte Applikation und Datenhaltung als Service anbietet. Die Benutzer können Informationen über gelesene Artikel bzw. Dokumente angeben. Dieser Content macht die Web-Plattform attraktiv für andere Benutzer, die über die Schlagwortsuche relevante Dokumente auffinden können. Wenn nun mehrere Benutzer bestimmte Themenbereiche als interessant erachten, entsteht eine Community. Das Funktionieren einer solchen Gemeinschaft hängt maßgeblich von der Ungezwungenheit und Freiwilligkeit des Informationsaustausches ab. Den Ausgangspunkt bilden dabei freie, kostenlose Services, die aber auch für kommerzielle Unternehmen interessant sein können, wie der nächste Abschnitt näher ausführt.

---

<sup>85</sup> Vgl. Gissing, Bernhard u. Tochtermann, Klaus (2007): „Corporate Web 2.0“, S. 27

<sup>86</sup> Vgl. studiVZ Ltd. (2008): „studiVZ“

<sup>87</sup> Vgl. Gissing, Bernhard u. Tochtermann, Klaus (2007): Ebenda, S. 16f

## 2.4.1 Unternehmen im Wandel

Die Inhalte und Beziehungen, die in Communities kreiert und gepflegt werden, sind vor allem privater Natur. Für Unternehmen stellt sich nun die Frage, ob und wie auf diese Tendenz reagiert werden soll. Ein Unternehmen, welches die Aufmerksamkeit der Bewegung rund um das Web 2.0 geschickt auf sich lenken konnte, ist Google. Services, wie u. a. Google Search (Suchmaschine), Google Mail (E-Mail), Google Maps (Landkarten) oder Google Earth (Satellitenbilder und Luftaufnahmen) stehen in erster Linie als freie Services zur Verfügung. Finanzierungsprobleme entstehen aber dadurch nicht, ganz im Gegenteil. Im Jahr 2006 konnte der Gewinn gegenüber dem Vorjahr um 250 Prozent auf 1,03 Mrd. Dollar gesteigert werden.<sup>88</sup> Der große Umsatz beruht auf dem Geschäftsmodell von Google, welches sich auf das Lukrieren von Werbeeinnahmen ausrichtet. Mit Google AdSense<sup>89</sup> können Anzeigen auf Webseiten platziert werden, die in Zusammenhang mit den Inhalten der Webseiten stehen. Im Falle der Google Suchmaschine werden auf bestimmte Schlagworte Werbebanner als Suchergebnisse angezeigt, deren Preis sich durch die Anzahl der erhaltenen Klicks errechnet.

Die Bildung von Communities zeichnet sich neben der Internetbranche auch in anderen Bereichen ab, die sich nicht ausschließlich mit dem Internet beschäftigen. Das Betriebssystem Linux<sup>90</sup> ist mit einer langen Liste von Distributionen, wie z. B. SuSE, Red Hat, Ubuntu etc., und einer im Jahre 2005 geschätzten Benutzerzahl von 29 Mio.<sup>91</sup> eine unumstrittene Erfolgsgeschichte von freier Software und Open Source Community Entwicklung. Viele Linux Versionen weisen halbjährliche Release-Zyklen auf, während die marktbeherrschende Konkurrenz Microsoft über fünf Jahre für die Entwicklung des Betriebssystems Windows Vista benötigt. Der erweckte Unmut bei den Anwendern ist nur allzu verständlich, wie eine im Mai 2007 durchgeführte Umfrage der Fachzeitschrift Tomorrow beweist. 69 Prozent der Befragten haben nämlich keine Ambitionen, auf Windows Vista umzusatteln.<sup>92</sup>

Die Einweg-Publikation von Webseiten wird in Zukunft nicht mehr ausreichen, um die gewünschte Zielgruppe anzusprechen. Firmen müssen ihre Kunden dazu ermutigen, ihre Produkte aktiv mit zu gestalten. Dies ist nicht nur auf Unternehmen der IT-Branche beschränkt. Der Spielzeughersteller Lego hat 2005 beschlossen, teure Legoprojekte mit einer

---

<sup>88</sup> Vgl. RP ONLINE GmbH (2007): „Google: Gewinn verdreifacht, Anleger enttäuscht“

<sup>89</sup> Vgl. Google (2008): „Google AdSense“

<sup>90</sup> Vgl. Linux Online (2008): „What is Linux“

<sup>91</sup> Vgl. The Linux Counter Project (2008): „Estimating the number of Linux users“

<sup>92</sup> Vgl. Bruckmeier, Jürgen (2007): „Wenig Lust auf Vista“

Community gemeinsam zu entwickeln. In dieser Gemeinschaft können erwachsene Lego Bauer ihre Erfahrungen austauschen. Aktive Teilnehmer in der Community werden mit bestimmten Zuwendungen bzw. Privilegien belohnt.<sup>93</sup> Die Kunst einer erfolgreichen Community ist ein Balanceakt: einerseits muss sich das Unternehmen öffnen und firmeninterne Informationen preisgeben, um Benutzer für Teilnahme und Kontributionen zu begeistern. Die entstandene Eigendynamik einer Community muss aber andererseits unter Kontrolle behalten werden, um einen Beitrag zur Erfüllung von strategischen Unternehmenszielen zu erbringen.

Die Möglichkeiten zur Bildung einer Community sind vielfältig und hängen nicht zuletzt vom Geschäftsfeld ab. Die nächsten beiden Abschnitte behandeln Ansätze, in der die soziale Interaktion bzw. die Bildung von Beziehungen im Vordergrund stehen.

## 2.4.2 Blogging

Der Begriff Blog ist eine Abkürzung des Begriffs Weblog und heißt übersetzt Webprotokoll oder Weblogbuch. Es beschreibt ein einsehbares Tagebuch oder Journal, welches chronologisch sortierte Beiträge in gestürzter Reihenfolge enthält, die in periodischen Abständen gegliedert werden. Die registrierten Benutzer können ihre Gedanken, Erfahrungen oder Information preisgeben, die zur eröffnenden Thematik passen. Blogging erfordert einfaches und schnelles Online-Editieren von Webseiten, dass in den Anfängen des WWW aufgrund folgender drei Gründe nicht möglich war:

- Das Erlernen der Fachsprache HTML war für die Masse nicht akzeptabel.
- Das Modifizieren der Seiten war umständlich (einwählen, einloggen, hochladen, kontrollieren etc.) und aufgrund der geringen Bandbreite war die Verbindung sehr langsam.
- Unternehmen wollten sich vor Verunstaltung, Angriffen und Missbrauch schützen.

Die Anzahl an Wikis und Foren, die als Austauschplattformen für beliebige Themenkomplexe bzw. Meinungen verstanden werden, nahmen im letzten Jahrzehnt deshalb exponential zu, weil das Posten eines Forumeintrags nicht mehr Wissen als das Schreiben einer Email

---

<sup>93</sup> Vgl. Gissing, Bernhard u. Tochtermann, Klaus (2007): „Corporate Web 2.0“, S. 20

erfordert. Ein Blog ist einem Forum sehr ähnlich und findet darüber hinaus in Literaturkreisen, so genannten LitBlogs, sowie bei traditionellen Printmedienhäusern, Anwendung. Der subtile Unterschied zwischen einem Blog und einem Forum lässt sich mit folgender Analogie erklären, mit der der Herausgeber des Blogs die Marschrichtung vorgibt (übrigens einem Blog entnommen):

„Blogs sind keine Gemeinschaftswohnungen. Es gibt einen Hausbewohner, den Gäste je nach Bedarf besuchen, weil er möglicherweise leckeren Kaffee und Kuchen anzubieten hat. Auf Foren und Newsgroups machen die Hausbewohner gemeinsam die Musik, jeder kann dazu seinen Kuchen mitbringen. Party! Auf einem Blog macht nur einer die Musik, er bestimmt die Lautstärke, er backt den Kuchen.“<sup>94</sup>

In der Recherche geprüfte Zeitungen, wie z. B. der Standard<sup>95</sup>, betreiben Blogs mit großem Erfolg, indem sie Artikel von ausgewählten Journalisten schreiben lassen. Jeder eingeloggte Leser kann einen Kommentar zum Artikel oder zu einem anderen Kommentar abgeben bzw. diesen bewerten. Die Journalisten lesen wiederum diese Beiträge und können damit Themen, die in den Blogs diskutiert wurden, wieder aufgreifen und für eigene Artikel verwenden. Die Qualität eines Blogs richtet sich natürlich nach der Kompetenz und Disziplin aller Beteiligten, so dass bei heftigen Kontroversen die Argumente nicht in abfälligen Bemerkungen enden sollten.<sup>96</sup>

Hyperlinks sind ein gutes Mittel, einen Beitrag mit externen Quellen zu belegen. Darüber hinaus können in einem Blog drei essentielle Funktionen bereit gestellt werden, die auf traditionellen Webseiten nicht vorzufinden sind:<sup>97</sup>

- *Permalinks*: Ein Eintrag erhält eine eigene Adresse, unter der er jederzeit aufgerufen werden kann. Die vermeintlich triviale Eigenschaft trägt zur gezielten Vernetzung bei.
- *Trackback*: Die Blog-Software wird informiert, wenn auf einen Eintrag im Blog Bezug genommen wird. Somit können die Einträge untereinander verlinkt werden.
- *Feeds*: Leser können bevorzugte Blogs abonnieren, um über neue Einstellungen informiert zu werden. Im Unterschied zu Newslettern bleibt der Nutzer anonym.

---

<sup>94</sup> Basic Thinking Blog (2005): „Unterschied Chat, Forum, Newsgroup, Blog“

<sup>95</sup> Vgl. Bronner Online AG (2007): „Standard.at“

<sup>96</sup> Der Autor spricht aus eigener Blog-Erfahrung

<sup>97</sup> Vgl. Alby, Tom (2007): „Web 2.0“, S. 22f

Ein Unternehmen kann durch die gezielte Einführung von Blogs sehr profitieren. Intern kann der Meinungs-austausch und die Wissensverteilung forciert werden, während extern auf Kundenwünsche schneller reagiert werden kann. Das Missbrauchsrisiko ist dementsprechend hoch, deshalb sollten Autorschaften und Benutzerrechte im Vorhinein festgelegt und zur Laufzeit überprüft werden, um dem Unternehmen unangenehme Beiträge zu vermeiden. Der nächste Abschnitt enthält nun andere Möglichkeiten, wie soziale Umgebungen im Web geschaffen werden können.

### **2.4.3 Social Software**

Der Begriff Social Software ist ebenso unscharf definiert wie der Begriff Web 2.0. Eines der Hauptziele ist der Aufbau, die Pflege und Selbstorganisation bzw. –regulierung einer Community. Chat und Instant Messaging Systeme (IM) fallen ebenfalls unter diesen Begriff. Im Laufe der Zeit haben sich zwei Kategorien herauskristallisiert:<sup>98</sup>

- Synchroner Kommunikation, die in der Regel nicht aufgezeichnet wird, steht im Vordergrund. Beispiele sind Voip Systeme, wie z. B. Skype oder VoipBuster, und IM Systeme, wie z. B. Internet Relay Chat (IRC), MSN Messenger oder ICQ
- Inhalte, die normalerweise asynchron in Form von Schrift, Bilder oder Videos kommuniziert werden, und der Community Gedanke sind wichtig. Beispiele hierfür sind die Enzyklopädie Wikipedia, das Fotoportal Flickr oder die Videosammlung YouTube.

Blogs fallen in die letztere Kategorie. Die folgende Mind Map versucht alle Typen zu erfassen, denen vor allem das Festhalten des Inhalts am Herzen liegt:

---

<sup>98</sup> Vgl. Alby, Tom (2007): „Web 2.0“, S. 89f

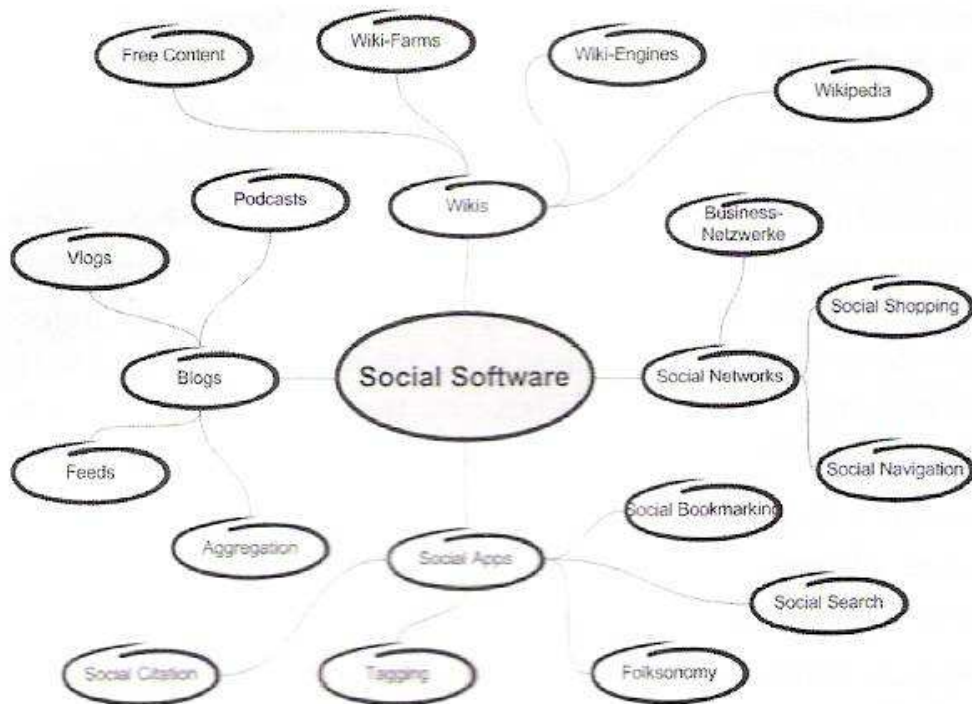


Abbildung 8: Mind Map Social Software<sup>99</sup>

Weitere Beispiele von Anwendungen sind del.icio.us (Social Bookmarking Dienst), Lycos IQ (Wissensgemeinde), last.fm (Musikempfehlungen), XING (Business Plattform), MySpace (Social Network) oder StudiVZ (Social Network). Social Software Systeme können virtuelle Dialoge erschaffen, die zum Knüpfen neuer Kontakte, Sammeln und Teilen von Wissen oder der alternativen Unterhaltung dienen können. Es sind immer die einfachen Werkzeuge mit komplexem Hintergrund, die Akzeptanz in der breiten Masse finden. Es bleibt abzuwarten, wann das Betreten eines Chat-Rooms so einfach wie das Abheben eines Telefonhörers ist. Ein kurzer Ausblick dazu folgt im nächsten Unterkapitel.

#### 2.4.4 Die Zukunft des Web

Die Zukunft des Webs hängt maßgeblich von den Akteuren ab. Die Internetfirma Google verschreibt sich in der Autobiographie „The Google Story“ folgender Philosophie:

„Unlike most companies, where executives and product managers try to think of ways to make money and then create products, Google is a place where technologists think first of ways to solve problems; only later, if ever, do they worry about how to ‚monetize‘ them.“<sup>100</sup>

<sup>99</sup> Szugat, Martin et al. (2006): „Social Software“, S. 18

Das Besondere am Geschäftsmodell von Google ist die Abwesenheit eines mittleren Managements. Projekte werden in Dreiergruppen abgewickelt, die die leitenden und organisatorischen Aufgaben wahrnehmen, und die volle Verantwortung der Konsequenzen übernehmen. Das Flair einer Community scheint bei Google schon etwas abgefärbt zu haben, und auch langfristig gesehen werden sich die Organisationsstrukturen von kommerziellen Unternehmen und freien Communities einander annähern. Ein Softwareprodukt, welches im Web erfolgreich sein will, wird an der Unterstützung einer oder mehrerer Communities nicht mehr vorbei kommen können. Das Bestehen eines Unternehmens wird auch maßgeblich davon abhängen, inwieweit die Organisation bereit ist, Kompetenzen an die Benutzer abzugeben.

Die virtuelle Realität hat bereits begonnen und die 3D Welt Second Life<sup>101</sup> bietet schon ein Vorgeschmack darauf, was uns in Zukunft erwartet. Alle Emotionen und Gefühle, die in der Realität Mangel erleiden und aufgrund von physischen Restriktionen bzw. gesellschaftlichen Konventionen nicht verwirklicht werden können, werden in diesen künstlichen Raum projiziert. Benutzer werden in die Rollen von Avataren schlüpfen, Studienkollegen an virtuellen Universitäten treffen, Sehenswürdigkeiten an weit entfernten Urlaubsorten begutachten, den Mount Everest besteigen, über Berge fliegen oder erotischen Fantasien freien Lauf lassen. Die virtuelle Welt wird eine wichtige, grafische Erweiterung des Webs sein, die als Plattform die bereits heute bekannten Funktionen der Partizipation, Benutzerbeiträge und Netzwerkinfrastruktur bietet.<sup>102</sup> Die große Gefahr besteht darin, dass die Konsequenzen der Naturgesetze nicht völlig vergessen werden. Das Begehen eines Fehlers in der realen Welt, wie z. B. dem Springen von einem Hochhaus, führt zu einem irreversiblen Schaden, der nicht einfach durch das Drücken der Reset Taste rückgängig gemacht werden kann.<sup>103</sup>

Die Visionen der virtuellen Realität werden noch einige Zeit brauchen, bis sie auf der Infrastruktur des Webs Fuß fassen können. Die Gegenwart beschäftigt sich u. a. mit Blogs, Podcasts (Tonsender), Wikis und Rich Internet Applikation (RIA), die erst den Sprung zur vereinnahmenden Multimedialität schaffen müssen. Ajax wird häufig in einem Atemzug mit RIAs genannt und als eine viel versprechende GUI Technologie des Web 2.0 eingestuft. Die

---

<sup>100</sup> Vise, David (2005): „The Google Story“, S. 6

<sup>101</sup> Linden Lab (2003): „Second Life“

<sup>102</sup> Vgl. Alby, Tom (2007): „Web 2.0“, S. 173

<sup>103</sup> Heute schmunzelt man vielleicht noch über solche Aussagen, die bei zufriedener Ausreifung der virtuellen Realität zu einem ernstem Problem werden können

Kombination von Web 2.0 Konzepten bzw. Methoden und das für viele Benutzer vertraute, desktop-ähnliche Aussehen bzw. Verhalten von Applikationen soll die Erfolgsaussichten von Ajax garantieren. Welche Fähigkeiten und Funktionalitäten Ajax zurzeit dafür vorweisen kann, wird im nächsten Kapitel genauer beschrieben.



### 3 Ajax

Dieses Kapitel klärt die Bedeutung des Begriffs „Ajax“. Zuerst soll das Konzept erläutert werden, dass nachher mit praktischen Anwendungen veranschaulicht werden soll. Es wird evaluiert, ob mit Ajax eine RIA konstruiert werden kann. Der verwendete Begriff Framework<sup>104</sup> wird im Rahmen dieser Arbeit als eine Umgebung verstanden, die neben Programmbibliotheken auch Funktionen zur Entwicklungsunterstützung bieten. Ein Framework befreit die Programmierung von Routineaufgaben mit Hilfe von vorgefertigten Standardlösungen und gewährt somit mehr Zeit für die eigentliche Entwicklung der Applikationen und den damit verbundenen Funktionen. Deshalb soll eine Reihe von Frameworks recherchiert werden, die zur Konstruktion von RIAs verwendet werden können. Danach werden anhand von eigens definierter Selektionsparameter die Frameworks bewertet. Es liegt nicht im Bestreben dieser Arbeit objektive Vor- und Nachteile der recherchierten Frameworks auszuarbeiten. Sie werden viel mehr auf die Kompatibilität mit ADOweb bewertet, d. h. welche Lösungen die Zielsetzung von ADOweb unterstützen. Als Ergebnis dieses Kapitels soll ein Framework bzw. eine Kombination dieser ausgewählt werden, die als Grundlage für die Implementierung des Prototyps für eine Ajax Literaturdatenbank dienen.

Der erste Abschnitt behandelt eine allgemeine Vorstellung von Ajax als Umsetzung einer RIA (Rich Internet Applikation), wie sie bereits in Abschnitt 2.2.4 angesprochen wurde. Es beinhaltet ferner einen Überblick über die eingesetzten Webtechnologien, die konzeptuelle Architektur, und die aktuellen Fähigkeiten bzw. Einschränkungen von Ajax. Der zweite Abschnitt listet zunächst eine Kategorisierung der mehr als 200 bekannten Frameworks auf, von denen etwas mehr als ein Dutzend einer genaueren Recherche unterzogen werden. Zur Beschreibung gehört ein grafisches Modell, das zeigen soll, wie das Framework mit einem Webservice, das die Schnittstelle von ADOweb repräsentiert, kommunizieren kann. Zusätzlich soll ein Beispiel einer Applikation, die mit diesem Framework bereits entwickelt wurde, vorgestellt werden, um so exemplarisch die Potentiale der Benutzeroberfläche im Webbrowser zu demonstrieren. Von den vorgestellten Frameworks wird im letzten Abschnitt eine Auswahl getroffen, die für die Umsetzung des Prototyps empfohlen wird. Dabei werden Selektionsparameter definiert, die als Hilfestellung bei der Entscheidungsfindung dienen sollen. Die Ergebnisse der Evaluierung bilden die Basis für die webbasierte Software, die im Kapitel 4 näher ausgeführt wird.

---

<sup>104</sup> Die wortwörtliche Übersetzung des Begriffs Framework ist Rahmenwerk

### 3.1 Das Konzept

Der Begriff „Ajax“ (Aynchronous JavaScript and XML) geht auf einen Anfang 2005 verfassten Aufsatz „Ajax: A New Approach to Web Applications“<sup>105</sup> von Jesse Garret zurück. Die Grundidee von Ajax besteht im Ausstatten des Webbrowsers mit einer intelligenten Darstellungs- und Sitzungsschicht (Ajax Engine), die den Inhalt und die Struktur von Webseiten verändern kann. Neben der Browser Add-On Programmiersprache JavaScript, die die Interaktivität erhöht, sind laut Garret die Technologien CSS und DOM wichtig. CSS kann zur Formatierung des Aussehens, wie z. B. Schriftgröße oder Hintergrundfarbe, eingesetzt werden. DOM hingegen bildet die Struktur von Webseiten ab und wird von JavaScript zum dynamischen Manipulieren von HTML Seiten verwendet.<sup>106</sup> Dadurch kann auf Ereignisse der Benutzeroberfläche, wie z. B. das Auffächern eines Verzeichnisbaums, clientseitig reagiert werden, ohne eine Anfrage an den Server stellen zu müssen. Das Nachladen von Information wird durch die Technik XMLHttpRequest, welches von Microsoft 1998 zum globalen Anbieten von Email- und Organizer Programmen für den Internet Explorer 5 entwickelt wurde,<sup>107</sup> bewerkstelligt. Dies erfolgt in der Regel asynchron, um die hohen Latenzzeiten des HTTP Verkehrs zu verbergen. Ajax setzt auf der Clientseite vorwiegend auf Technologien, die von gängigen Browsers standardmäßig unterstützt werden. Deshalb werden in vielen Ajax Frameworks häufig auch Cookies, DHTML oder XSLT eingesetzt, während auf die Einbindung von Plugins, wie z. B. Flash, eher verzichtet wird. Die Bindung an eine bestimmte Technologie ist jedoch bei diesem Konzept nicht zwingend vorgeschrieben, weshalb sich eine Ajax Applikation allgemein definieren lässt:

„An Ajax application builds on standard web technologies to deliver a rich, responsive user experience“<sup>108</sup>

Die wichtigste Aufgabe des Servers, welcher durch eine beliebige Programmiersprache wie Java, PHP, C++, Perl oder Python implementiert werden kann, ist das Bereitstellen von Information im XML Format, die im Gegensatz zur HTML Dokumenten keine Darstellungselemente enthalten. Die architektonischen Unterschiede einer herkömmlichen Webapplikation gegenüber einer Ajax Webapplikation zeigt die folgende Abbildung:

---

<sup>105</sup> Vgl. Garret, Jesse (2005): „Ajax: A New Approach to Web Applications“

<sup>106</sup> Vgl. Flanagan, David (2001): „JavaScript: The Definitive Guide“, In: O'Reilly & Associates, K. 17 (The Document Object Model)

<sup>107</sup> Vgl. Denny Carl (2006): „Praxiswissen Ajax“, S. 6

<sup>108</sup> Mahemoff, Michael (2006): „Ajax Design Patterns“, S. 6

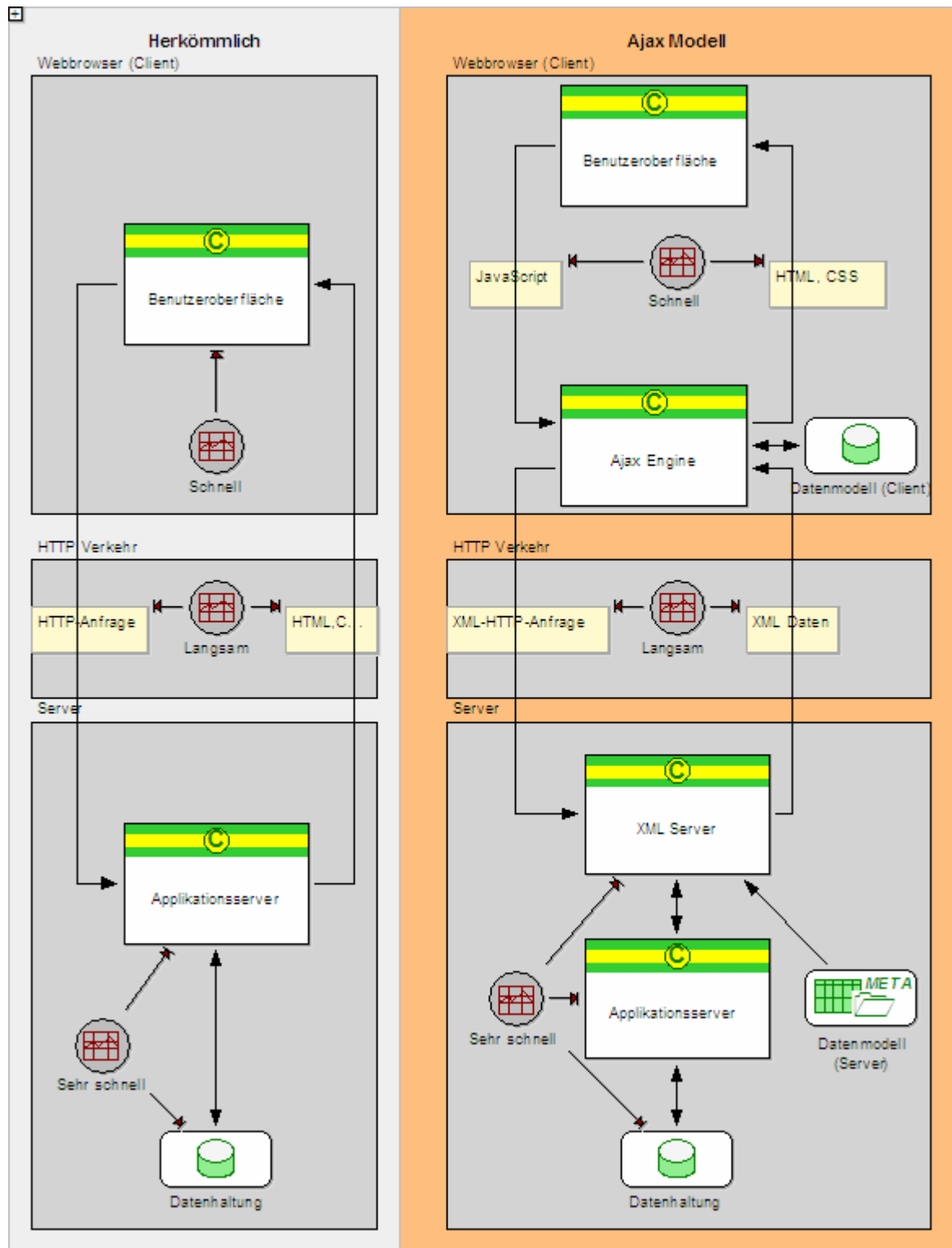


Abbildung 9: Herkömmliche vs. Ajax Webapplikation

Ajax Applikationen erweitern das traditionelle Webanwendungen Modell um zwei Komponenten, nämlich der Ajax Engine und den XML Server. Die Ajax Engine beinhaltet hauptsächlich GUI Logik, wie Widgets<sup>109</sup> und dessen Ereignisse<sup>110</sup>, um auf Benutzerbefehle

<sup>109</sup> Benutzeroberflächenelemente, wie Schaltflächen, Tabellen, Bäume, Textfelder, Labels etc.

<sup>110</sup> Mausgestiken, wie Klicken, Drag and Drop, Markieren bzw. Tastatureingaben

schneller reagieren zu können. Optional kann diese Engine mit einem Datenmodell ausgestattet werden, dessen Inhalt vom XML Server gesteuert bzw. synchronisiert wird. Der Zweck dieser Erweiterungen besteht darin, den Engpass des HTTP Verkehrs mit Hilfe der schnelleren, lokalen Server- bzw. Clientverarbeitung zu überbrücken. Dies wird erreicht durch die Verlagerung der Benutzeroberflächenverwaltung vom Server auf den Browser.<sup>111</sup> Die Aufgaben des Servers werden auf die Lieferung bzw. den Empfang von Daten beschränkt, die in unabhängige Funktionen gepackt, wie z. B. in Form von Webservices, angeboten werden können. Ajax kann für eine Reihe von Anwendungen, wie z. B. Buchhaltungs- oder Abrechnungsprogramme, Zeitplanungssysteme und sogar Office Applikation wie MS Word oder Excel, als äquivalenter Ersatz dienen. Die Nachbildung von Applikation mit einem hohen Grad an grafischer Manipulation, wie z. B. Corel Draw oder ADONIS, ist aufgrund jüngster JavaScript Bibliotheken, wie z. B. Draw2D<sup>112</sup> von OpenJacob, und die Unterstützung von SVG<sup>113</sup> Grafiken zumindest möglich geworden. Zur Abschätzung der aktuellen Möglichkeiten ist es hilfreich zu wissen, was Ajax zurzeit noch nicht unterstützt:<sup>114</sup>

- *Browser morphing*: Das Hinzufügen von Menu- bzw. Toolbareinträge, das Verwalten von Lesezeichen und das Reagieren auf Browser Verhalten ist nicht möglich.
- *Zugriff auf lokale Dateien*: Es existieren keine Lese- bzw. Schreibrecht von Dateien, die sich auf der Festplatte des Benutzers befinden. Eine Ausnahme sind File Uploads oder Cookies, die sehr begrenzte Möglichkeiten aufweisen.
- *Ton Effekte*: Das Abspielen von Musik oder Ton wird kaum unterstützt. Dies wird meist durch eingebettete Plugins realisiert, wie z. B. RealPlayer oder MPlayer.
- *Reichhaltige Grafik*: Das Anbieten von komplexen Grafiken, die sich dynamisch und schnell ändern, steckt noch in den Kinderschuhen.
- *Keyboard Shortcuts*: Browser Shortcuts können Ajax Shortcuts überdecken und rufen somit Konflikte hervor.
- *Zugriff auf die Hardware*: Die Eingabe über Mikrophone und Webcams oder die Ausgabe auf einen Drucker werden nicht unterstützt.
- *Erweiterte Kommunikation*: Die Kommunikation geht nicht über die Client-Server Unterhaltung und das HTTP Protokoll hinaus.

---

<sup>111</sup> Vgl. Rauschmayer, Axel (2006): „Ajax mit DWR und Dojo“, S. 3

<sup>112</sup> Vgl. OpenJacob (2007): „Draw 2D“

<sup>113</sup> Vgl. W3C (2007): „Scalable Vector Graphics (SVG)“

<sup>114</sup> Vgl. Charland, Andre (2007): „Will AJAX Replace the Desktop?“

- *Interaktion mit dem Betriebssystem:* Reagieren auf Ereignisse des Betriebssystems, das Ändern der Systemeinstellungen oder das Lesen von Hardware Information ist nicht möglich.

Diese Einschränkungen tun der wachsenden Popularität von Ajax keinen Abbruch. Ajax punktet mit folgenden Vorzügen:<sup>115</sup>

- *Applikationen anstatt statischer Webseiten:* Das WWW ist nicht länger eine lose Dokumentensammlung, sondern kann zur Etablierung einer Plattform genutzt werden, die ein desktop-ähnliches Look & Feel aufweist.
- *Flüssige Interaktion:* JavaScript baut die Benutzeroberfläche auf, fängt Ereignisse ab und sendet Anfragen im Hintergrund an den Server (XMLHTTP). Diese Maschinerie gewährleistet einen hohen Grad an Rückkopplung (Feedback) und Interaktivität.
- *Live:* Webseiten werden ständig mit frischen Inhalten aktualisiert, ohne das komplette Dokumente nachgeladen werden müssen.
- *Unterstützend:* Ajax hilft den Benutzern bei der Arbeit, wie z. B. durch Autovervollständigung von Suchbegriffen.
- *Visuelle Effekte:* Grafische Animationen, wie z. B. skalierbare Bilder (SVG), verschönern die Benutzeroberfläche bzw. erleichtern die Bedienung
- *Zusätzliche Widgets:* Benutzeroberflächenelemente, wie z. B. Schieberegler oder Fortschrittsanzeige, die in HTML nicht möglich sind, werden unterstützt.
- *Neue Stile der Interaktion:* Neue Mausgestiken, wie Drag und Drop, Keyboard Shortcuts, sowie die Kombination beider, werden unterstützt.
- *Standard basiert:* Ein gängiger Browser mit aktiviertem JavaScript und ein eventuelles Erlauben von Cookies sind ausreichend.

Ajax Frameworks bieten Unterstützung für bestimmte Funktionen an, die sich je nach Ausrichtung besser oder schlechter für die Integration in ADOweb eignen. Der nächste Abschnitt erläutert diesen Aspekt anhand von konkreten Beispielen.

---

<sup>115</sup> Vgl. Mahemoff, Michael (2006): „Ajax Design Patterns“, S. 10

## 3.2 Übersicht und Vergleich Ajax Frameworks

Dieser Abschnitt stellt die Resultate einer Recherche vor, die ausgewählte Open Source Ajax Frameworks analysiert. Das Angebot an verfügbaren Lösungen ist sehr umfangreich, weshalb sich die Vorgehensweise der Recherche in drei Schritten gliedert

- Suche bzw. Vorschau im Internet
- Vergleich mit Hilfe von Parametern
- Diskussion und Schlussfolgerung des Szenarios

Die erste Bemühung besteht darin, sich einen Überblick über das Angebot von Ajax Frameworks zu verschaffen. Im Zuge dessen wird man auf der Seite „[ajaxpatterns.org](http://ajaxpatterns.org)“ fündig, die 219 verschiedene Frameworks auflistet und beschreibt. Dort erfolgt eine Kategorisierung in JavaScript, serverseitigen und hybriden Frameworks, dessen weitere Unterteilungen im Folgenden kurz aufgelistet werden. Zu den JavaScript Frameworks, die die Darstellung und Interaktion im Browser unterstützen, zählen (in den Klammern steht die Anzahl der Frameworks):<sup>116</sup>

- *Multipurpose frameworks (39)*: Widget Bibliotheken, Ereignisverarbeitung, eventuelles Datenmodell und teilweises Kapseln der Kommunikation zum Server
- *Remoting Frameworks (20)*: Clientseitige Kommunikationslogik mit Hilfe von XMLHTTP Anfragen und Stubs (Adapter)
- *Effects Frameworks (8)*: Nette, visuelle Effekte im Browser, wie z. B. Drag & Drop
- *Flash Frameworks (3)*: Multimediales Flash, dass von JavaScript oder einer anderen Scriptsprache, wie z. B. ActiveX, dynamisch generiert wird
- *Logging Frameworks (5)*: Zusätzlich Komponenten für Logging und Monitoring
- *XML Frameworks (8)*: Verwendung von XML Werkzeugen, wie z. B. XHTML oder XSLT zur Darstellung bzw. Transformation von Information
- *Specialised Frameworks (3)*: Sonstige Frameworks, die sich auf bestimmte Teilgebiete spezialisieren, wie z. B. Image Viewer (Bildanzeige), Drag-Drop oder Auto-complete (Vervollständigen von Suchen bzw. Texteingaben)

---

<sup>116</sup> Vgl. Mahemoff, Michael (2007): „Ajax Patterns“

Serverseitige bzw. hybride Frameworks unterstützen zusätzlich die Akquisition bzw. Aggregation von Information mit Hilfe von objektorientierten bzw. scriptbasierten Programmiersprachen, die darüber hinaus die Applikationslogik übersichtlicher und leichter wartbar machen:

- C++ Ajax Frameworks (1)
- Coldfusion (4)
- DotNet (21)
- Java (46)
- Lisp (1)
- Lotus Notes (0)
- Multi-Language (11)
- Perl (2)
- PHP (39)
- Python (5)
- Ruby (1)
- SmallTalk (2)

Ein wichtiger Maßstab bei der Bewertung der Frameworks ist die Unterstützung durch eine Community. Deshalb wurde im Jahr 2006 von der renommierten Ajax News Seite „Ajaxian.com“ eine Umfrage durchgeführt, die zum Ziel hatte, den Beliebtheits- und Verbreitungsgrad von Ajax Frameworks herauszufinden. An der Umfrage nahmen 865 Personen teil, die Ajax hauptsächlich für folgende Zwecke in Betracht ziehen<sup>117</sup>:

- Produktion: 61,7 %
- Entwicklung: 67,2 %
- Prüfen des Konzepts: 33,5 %

Ajax ist demnach über die Evaluierungsphase hinweg, in der es als vager Gegenstand der Forschung eingestuft wird, dessen gegenwärtige bzw. zukünftige Existenzberechtigung noch in den Sternen steht. Das Konzept findet bereits seinen Platz in der alltäglichen Welt des WWW. Die beiden, folgenden Grafiken geben die Auswertungen nach Distribution und Plattform wieder:

---

<sup>117</sup> Almaer, Dion et al (2006): „Ajaxian.com 2006 Survey Results“

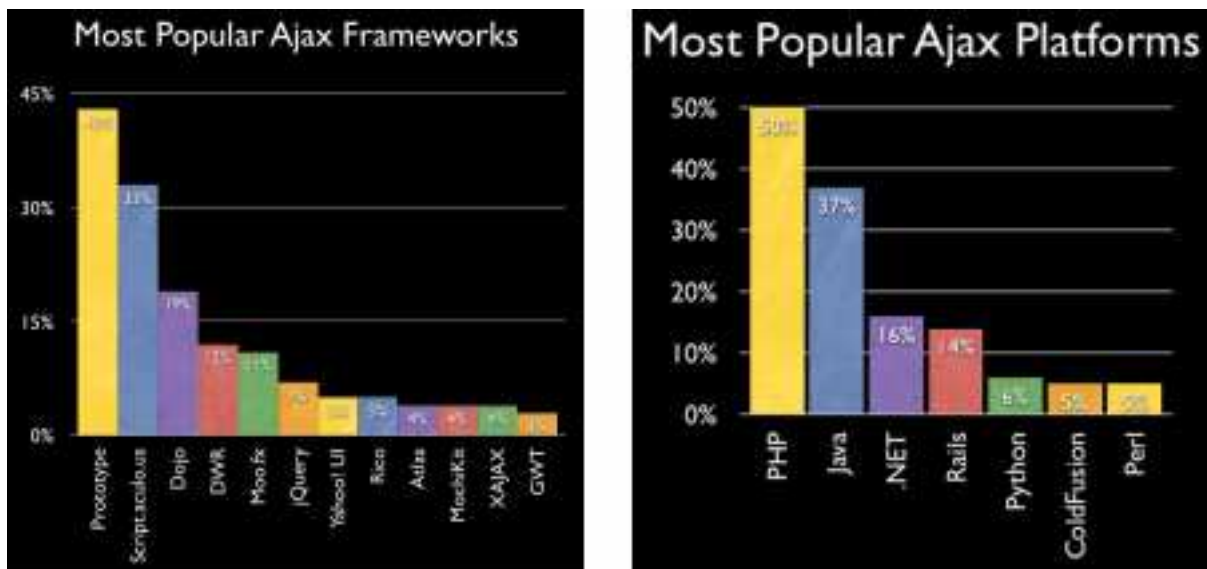


Abbildung 10: Umfrage über Ajax Frameworks und Plattformen<sup>118</sup>

Das Low-level Framework Prototype, welches sich mit Kommunikations- und Verbindungsdetails beschäftigt, hat den größten Marktanteil. Danach folgt das Effekt Framework Script.aculo.us und das Mehrzweck (Multipurpose) Framework Dojo. Die dominierende Plattformsprache ist PHP, gefolgt von Java und .NET. Die Studie muss sich aber folgende Kritik gefallen lassen:

- Die Frameworks sind kaum vergleichbar, da große Unterschiede in der Funktionalität und Ausrichtung bestehen.
- Low-level Frameworks werden oft in High-level Frameworks integriert bzw. kombiniert. Deshalb steht Prototype an erster und GWT an letzter Stelle.
- Die Umfrage ist aufgrund des Ausmaßes der Beteiligung (865 Personen) nur begrenzt aussagekräftig und schließt nicht aus, dass es sich um eine bestimmte Lesergruppe von „Ajaxian.com“ handelt.
- Low-level Frameworks können leicht für kleinere Projekte verwendet werden, von denen eine Vielzahl existieren. High-level Frameworks werden nur in wenigen, großen Projekten verwendet.

Die Studie ist ein guter Startpunkt. Nun wird eine selektive Vorauswahl von Frameworks getroffen, die sich auf Information von ajaxpattern.org, ajaxian.com, Internet Recherche,

<sup>118</sup> Almaer, Dion et al (2006): Ebenda



Fachbüchern und persönlicher Erfahrung stützen. Relevante Framework Kandidaten auf Basis der Umfrage sind:

- Prototype
- Script.aculo.us
- Dojo
- DWR
- jQuery
- Yahoo! UI
- Rico
- Atlas (ASP.NET)
- Xajax
- GWT

Im Zuge einer erweiterten Internet Recherche machten noch andere Frameworks auf sich aufmerksam, die aufgrund außergewöhnlicher Funktionalitäten auch in die Analyse mit einbezogen werden sollen. Die Auswahl stützt sich auf Beschreibungen und Demoapplikationen:

- ThinWire
- OpenJacob
- Qooxdoo
- RAP (Rich Ajax Platform)

Die nächsten 14 Abschnitte stellen die Frameworks etwas detaillierter vor. Neben einer kurzen Beschreibung erläutert ein grafisches Modell die Integration mit einem beliebigen Webservice, das in diesem Fall die Schnittstelle zu ADOweb repräsentiert. Abschließend wird beispielhaft eine Applikation aus dem Framework vorgestellt, um einen Eindruck von der für die Endbenutzer zur Verfügung gestellten Benutzeroberfläche zu erhalten.



### 3.2.1 Prototype

Bei Prototype<sup>119</sup> handelt es sich laut der Umfrage von Ajaxian.com um das beliebteste Ajax Framework. Es ist clientseitig und mehrzweckfähig und bietet low-level Unterstützung für Ajax Entwicklung, welches XMLHttpRequest Anfragen kapselt und eine erweiterte DOM (Data Object Modelling) Manipulation von XML Strukturen ermöglicht. Prototype ist ein im Jahr 2005 gegründetes Spin-Off des „Ruby on Rails“ Projekts und hält bei der Version 1.6 fest. Die Kern Community zählt über 3.000 Interessierte Blog Teilnehmer und wird von anderen Communities wie Script.aculo.us, Rico oder Ruby oft referenziert, da diese Prototype als Basis Bibliothek nutzen. Die verwendeten Technologien sind HTML, JavaScript und JSON Features.

Der Hauptgrund für den Erfolg dieses Frameworks liegt in der einfachen Integrierung in andere Projekte, da die Bibliothek klein und effizient ist. Die Entwicklung erfordert die manuelle Generierung von HTML Seiten, die Verweise auf externe JavaScript Dateien haben. Die Client Server Kommunikation ist in diversen Methoden der JavaScript Bibliotheken, wie z. B. Ajax.Request, Ajax.Updater oder Ajax.Responders, gekapselt. Dabei handelt es sich um GET und POST Methoden des HTTP Protokoll, die mit semantischer Funktionalität, wie z. B. das Aktualisieren von Teile einer Seite, erweitert werden. Das Einbinden erweiterter Bibliotheken, wie z. B. ws.js<sup>120</sup>, welche zum Aufruf von SOAP Webservices verwendet wird, erfolgt über denselben Referenzeintrag in eine HTML Datei wie die internen JavaScript Bibliotheken. Eine Prototype Applikation verwendet nun diese JavaScript API für die eigene Logik, welche dem Client statisch übersendet wird. Die Kommunikation mit einem externen Webservice erfolgt entweder über die ws.js Bibliothek oder über einen eigenen Umschlag (Wrapper), wie es die folgende Grafik zeigt:

---

<sup>119</sup> Vgl. Prototype Core Team (2007): „Prototype JavaScript Framework“

<sup>120</sup> Vgl. Snell, James (2007): „Call SOAP Web services with Ajax ,Part 2: Extending the Web service client“

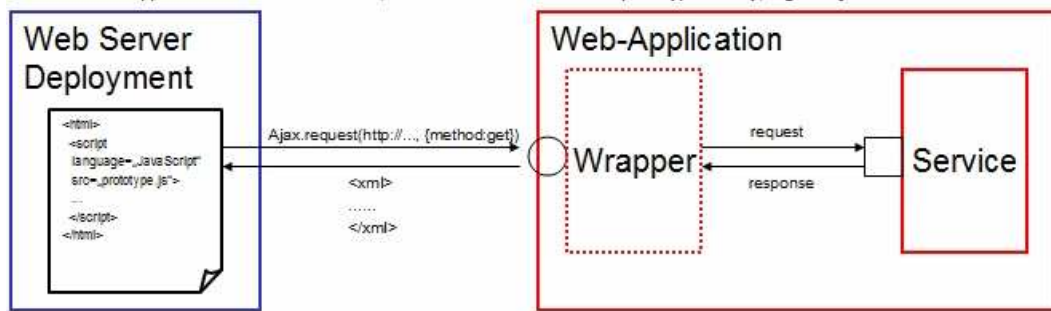


Abbildung 11: Prototype Architektur mit externem Service

Entwicklungsunterstützung bieten gewöhnliche JavaScript und HTML Editoren. Der Einsatz auf einem Webserver ist einfach, da die Seiten nur in das korrekte Unterverzeichnis kopiert werden müssen. Dieses Framework ist auch Thema von zahlreichen Konferenzen, wie z. B. von „The international PHP conference 2006“ oder „The Ajax Experience San Francisco 2007“. Es gibt keine direkten Kooperationen mit anderen Firmen, jedoch wird Prototype von Unternehmen, wie Apple, NBC, Amazon, H&M, SONY oder CNN genutzt.<sup>121</sup> Das Framework ist ziemlich ausgereift in Bezug auf Kommunikation, DOM-Manipulation (Dynamisches Ändern der Struktur von HTML Inhalten) und Dokumentation, jedoch wird die Entwicklung von RIAs nicht direkt ermöglicht. Des Weiteren zeichnet sich Prototype durch eine gute Browserunterstützung aus.

Die folgende Abbildung dient zur Veranschaulichung der Funktionalität von Prototype. Das GUI besteht aus einfachem HTML Formularcode, während Prototype im Hintergrund Postleitzahlen mit Hilfe eines serverseitigen Datenmodells verifiziert:

So the user first sees this:

zip:

Once the zip code is entered, and the response received it looks like this:

zip:  ✔ Utica, NY

If the zip code is not found in the db:

zip:  Zip code (00001) not found.

Abbildung 12: Prototype Postleitzahl Überprüfung<sup>122</sup>

<sup>121</sup> Vgl. Prototype Core Team (2007): „Who’s using Prototype“

<sup>122</sup> Freitag, Pete (2005): „Ajax Tutorial with Prototype“

Das Framework kann als durchaus geeignet im Hinblick auf eine Kombination ADOweb bewertet werden. Für kleinere Projekte ist es ausreichend. Applikationen, die zusätzliche GUI Funktionalitäten benötigen, wie z. B. die Literaturdatenbank, ist Prototype nur in Kombination mit einem anderen Framework sinnvoll.

### 3.2.2 Script.aculo.us



Bei Script.aculo.us<sup>123</sup> handelt es sich wiederum um ein clientseitiges Mehrzweck Framework. Im Unterschied zu Prototype konzentriert sich Script.aculo.us auf die Anzeige und Interaktion im Browser. Dazu baut Script.aculo.us auf den Diensten von Prototype auf und bietet visuelle Effekte, Drag & Drop und Unit Testing. Script.aculo.us wird seit 2005 entwickelt und ist bei der Version 1.8.1. Es ist ein Open Source Projekt des „Fluxion“ Service der Wollzelle GmbH. Die dazugehörige Google Group „Ruby on Rails: Spin-Offs“ zählt mehr als 4.000 registrierte Mitglieder und hat starke Verbindungen zu anderen Communities, allen voran Prototype. Wenn man es ohne Prototype betrachtet, handelt es eigentlich um eine DHTML Bibliothek, die mit Hilfe von Prototype Ajax fähig gemacht wird.<sup>124</sup> Die verwendeten Technologien sind HTML, DHTML, JavaScript und Prototype.

Script.aculo.us ist eine Bibliothek von JavaScript Objekten, ohne eine Bindung an einen serverseitigen Service. Die HTTP Kommunikation wird von Prototype erledigt, welches von der Server Plattform, wie z. B. Ruby on Rails oder C++, integriert wird. Für Java ergeben sich u. a. folgende Integrationsmöglichkeiten:

- *Ajax Tags*: Die Ajax Tag Library ist ein Satz von JSP (Java Server Pages) Tags, die die Verwendung von AJAX Technologie in Java Server Pages vereinfachen.
- *AppFuse*: AppFuse ist eine Applikation für schnelle Webapplikationsentwicklung. Hibernate oder iBATIS können als persistente, während JSF, Spring MVC, Struts, Tapestry oder WebWork als Web Frameworks verwendet werden. AppFuse liefert eine Konfiguration für Script.aculo.us mit.

---

<sup>123</sup> Vgl. Wollzelle GmbH (2007): „Script.aculo.us“

<sup>124</sup> Vgl. Mintert, Stefan und Leisegang, Christoph (2007): „Ajax“, S. 127f

Die folgende Abbildung zeigt eine Kombination von Script.aculo.us und Prototype mit JSF (Java Server Faces) oder den JSP basierten Ajax Tags:

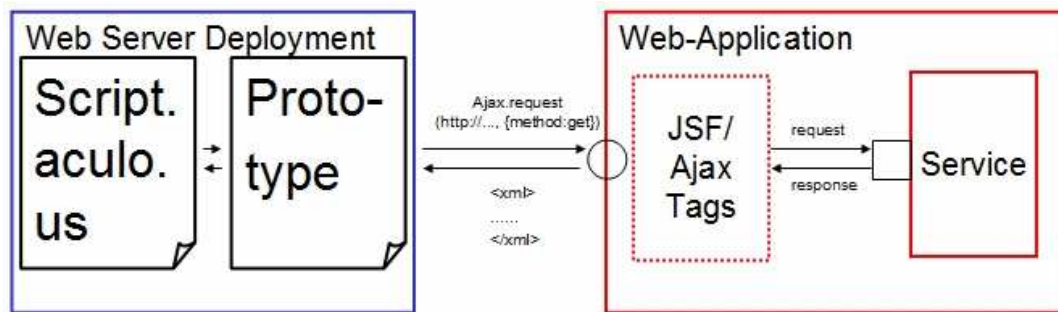


Abbildung 13: Script.aculo.us Architektur

Script.aculo.us empfiehlt sich für visuelle Effekte, die als netter Zusatz einer Webseite eingebaut werden können. Ein typisches Feature zeigt die folgende Abbildung, die ein Drag und Drop Szenario darstellt, welches Elemente in eine Liste einfügen kann:

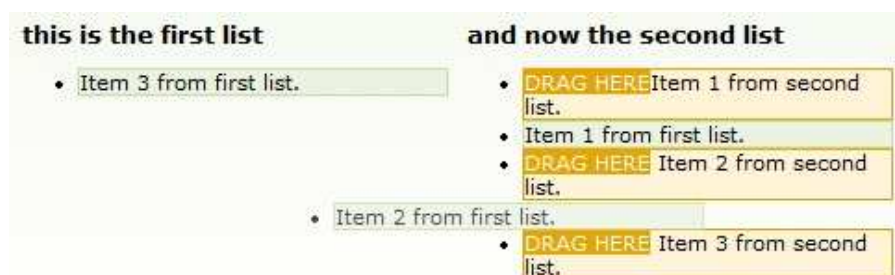


Abbildung 14: Script.aculo.us Drag und Drop Demo

Für die Realisierung einer ADOweb Rich Internet Applikation müsste ein großer Entwicklungsaufwand betrieben werden, da viele Basisfunktionalitäten, wie z. B. gewöhnliche Widgets, selbst implementiert werden müssen. Zur Reduzierung des Aufwands könnten weitere Frameworks, wie DWR, YUI, Dojo oder Tacos, im „Web Server Deployment“ inkludiert werden, um die Realisierung einer desktop-ähnlichen Umgebung zu verwirklichen.

### 3.2.3 Dojo



Dojo<sup>125</sup> ordnet sich wie Prototype und Script.aculo.us in die Kategorie der clientseitigen Mehrzweck Frameworks ein. Es handelt sich dabei um einen Open Source DHTML JavaScript Toolkit. Browserabhängigkeiten werden durch angepasste Skripte von einer schlanken und effizienten Bibliothek gekapselt. Darüber hinaus bietet Dojo eine mächtige Bibliothek an Widgets, die eine intuitive Gestaltung der Benutzeroberfläche erlauben. Das spannendste Feature von Dojo ist wahrscheinlich die Manipulation von SVG Grafiken, die skaliert, vergrößert bzw. verkleinert werden können. Die nächste Abbildung zeigt diese Eigenschaft anhand einer Dockbar, dessen Elemente bei Mausberührung aus der Liste springen:



**Abbildung 15: Dojo Fisheye Demo mit skalierbaren SVG Grafiken**

Das Framework wird seit 2005 entwickelt und befindet sich mit der Version 1.0.2 in einer stabilen Phase. Der Name Dojo taucht in renommierten Ajax Magazinen, wie z. B. der Ajaxian oder Golem, auf und ist auch in Entwicklerkreisen nicht unbekannt. Es herrscht eine rege Beteiligung im Forum von Dojo, was gute, zukünftige Aussichten dieses Projekts verspricht. Die Technologien stützen sich auf DHTML, SVG, JavaScript und HTML.

Dojo kann als Transformationsservice für das Front End eingesetzt werden. Die Anbindung an einen serverseitigen Service wird mit der Funktion `dojo.io.bind` zwar unterstützt, jedoch bietet sich die Kombination mit einem anderen Framework, wie z. B. DWR, an, welches dafür effizientere und mächtigere Mechanismen aufweist.

---

<sup>125</sup> Vgl. Dojo Foundation (2007): „Dojo – The JavaScript Toolkit“

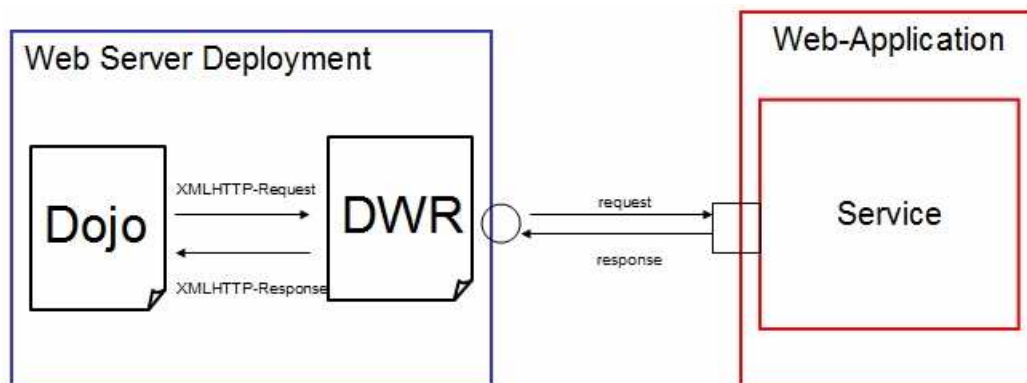


Abbildung 16: Dojo/DWR Architektur

Als Entwicklungsunterstützung bieten sich standardmäßige JavaScript bzw. HTML Editoren, sowie das Eclipse ATF<sup>126</sup> (Ajax Toolkit Framework), welches eine Editierungs- und Kompilierungskomponente bietet, an. Die Dokumentation ist ausreichend und das Framework hat trotz seiner Beta Versionen einen akzeptablen Reifegrad erreicht.

Für ADOweb ist neben der Widget Bibliothek vor allem die Manipulation von SVG Grafiken interessant, die einen hohen Grad an dynamischer, grafischer Manipulation erlaubt. Dojo stellt für ADOweb eine gute Alternative dar.

### 3.2.4 DWR



DWR<sup>127</sup> (Direct Web Remoting) ist ein serverseitiges Mehrzweck Framework, welches einen optimalen Einstieg für Java Entwickler in die Welt von Ajax bietet. Bei DWR handelt es sich um eine Java zu JavaScript Kompilierungsmaschine, die die Komplexität der Client-Server Kommunikation verbirgt. Java Methoden werden in äquivalente JavaScript Methoden übersetzt. Die dynamisch generierten JavaScript Dateien senden wiederum Anfragen an ein Java Servlet, welches die Nachrichten richtig interpretieren kann. Als Parameter können primitive Typen, wie z. B. Strings, Zahlen oder ein Datum, sowie auch DOM Objekte, Arrays und Collections übergeben werden. DWR wird seit 2005 entwickelt und hat mit der aktuellen Version 2.0 ein stabiles und reifes Release, welches asynchrone Aufrufe des Client unterstützt (Reverse Ajax).<sup>128</sup> DWR wird in der „SpringOne 2006“ Konferenz ebenso thematisiert, wie in

<sup>126</sup> Vgl. Eclipse Foundation (2007): „ATF (AJAX Toolkit Framework)“

<sup>127</sup> Vgl. Getahead (2007): „Direct Web Remoting (DWR)“

<sup>128</sup> Vgl. Mintert, Stefan u. Leisegang, Christoph (2007): „Ajax“, S. 170

den Blogs zahlreicher anderer, clientseitiger Frameworks. Es ist sehr beliebt, weil es leicht in bestehende Java Servlet Umgebungen integriert werden kann. Unternehmen wie Getahead oder TIBCO unterstützen die Entwicklung dieses Frameworks, und stellen eine Dokumentation bereit, die aus einem Online Manual und einigen einführenden Beispielen besteht. Die verwendeten Technologien sind Java, JavaScript und HTML.

DWR kann optimal als Verbindungsglied für Java betriebene Servlet Container und einen Ajax JavaScript Toolkit genutzt werden. Externe Frameworks, wie z. B. das java-basierte ADOweb Framework, können in die Verarbeitungsroutine eingebunden werden. DWR bietet keine eigene Widget Bibliothek an, womit auf HTML zurückgegriffen werden müsste. Deshalb wird DWR gern mit Dojo oder jQuery kombiniert, um ein besseres Aussehen und eine höhere Effizienz in der Entwicklung zu gewährleisten. Daraus ergibt sich eine kombinierte Architektur, wie sie bereits im Kapitel über Dojo (siehe Abschnitt 3.2.3) vorgestellt wurde. Zur Erstellung des serverseitigen Java Codes bieten sich Entwicklungsumgebungen wie Netbeans oder Eclipse an. Die Darstellung auf der Clientseite muss dennoch manuell implementiert werden, wie die nächste Abbildung anhand eines einfachen HTML basierten Tabelleneditor zeigt:

**All People**

Person	Salary	Actions
Fred Wilson 96 Brown Street, San Mateo	\$81000	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Leanne Nield 5 Green Avenue, New York	\$42000	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Martha Cowan 25 Green Street, Paris	\$71000	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Matt Daniels 68 Black Street, Kettering	\$21000	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

**Edit Person**

Name:

Salary:

Address:

(ID=1216)

**Abbildung 17: DWR Tabellen Editor**

Die Oberfläche soll jedoch nicht über die Tatsache hinweg täuschen, dass die eigentlichen Vorzüge des Frameworks in der Synchronisation mit dem Server liegt. DWR in Kombination



mit Dojo empfiehlt sich aufgrund der guten Integrations- und Entwicklungsmöglichkeiten mit ADOweb.

### 3.2.5 jQuery



Bei jQuery<sup>129</sup> handelt es sich um ein clientseitiges JavaScript Mehrzweck Framework. Im Prinzip ist es eine JavaScript Abfragebibliothek, die sich auf DOM Parsing bzw. Manipulation konzentriert. Eine neuere Entwicklung ist jQuery UI, welches Widgets zum Bau von Benutzeroberflächen zur Verfügung stellt. jQuery wird seit 2005 entwickelt und die neueste Version ist 1.2.3. Das eigentliche Entwicklungsteam besteht aus europäischen und amerikanischen Entwicklungsteams. jQuery basiert auf JavaScript, HTML und CSS.

Die Architektur und die Integration mit ADOweb sind analog zu Prototype (siehe 3.2.1). jQuery wird als einzelne JavaScript Datei ausgeliefert. Plugins werden wie dessen Kern mittels einer Referenz in der HTML Datei inkludiert. Für die Entwicklung sind Details über die Schnittstellen interessant, die verschiedenste Code Fragmente andockbar machen. Es existieren auch Erweiterungen für visuelle Effekte und das Front End (jQuery UI), die gut eingesetzt werden können. Die Unterstützung der Entwicklung gewährleisten Standard JavaScript und HTML Editoren. Dokumentation, Tutorials und eine API gewährleisten zudem einen guten Einblick in das Framework. Die folgende Abbildung zeigt in der linken Spalte diverse Funktionen, die den Inhalt von HTML Seiten auf Benutzerwunsch verändern können:

The image shows a screenshot of the jQuery HTML Editor interface. On the left, there are five examples (A through E) with interactive buttons and descriptions. Example A: '# of Paragraphs' button, 'Show / Hide jquery code'. Example B: 'Slide Out' and 'Slide In' buttons, 'Show / Hide jquery code'. Example C: 'Add' and 'Remove' buttons, 'Show / Hide jquery code'. Example D: 'Remove' button, 'Show / Hide jquery code'. Example E: 'Remove' button, 'Show / Hide jquery code'. On the right, there is a code editor area titled 'Column 2:'. It contains a red header box with the text 'USE THE BUTTONS TO THE LEFT TO RUN JQUERY CODE ON THE HTML BELOW.' Below this, there is placeholder text: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue dui dolore te feugiat nulla facilisi. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam,'

Abbildung 18: jQuery HTML Editor

<sup>129</sup> Vgl. Resig, John (2007): „jQuery“

jQuery in Kombination mit jQuery UI ist für ADOweb eine äquivalente Alternative zu Dojo. Sie unterscheiden sich etwas im Aussehen der Widgets. Die endgültige Wahl ist dem Geschmack der Entwicklung überlassen, die in etwa zwei vergleichbar gute Frameworks vorfinden sollten.

### 3.2.6 Yahoo! UI (YUI)



Auch Yahoo<sup>130</sup> bietet ein clientseitiges Mehrzweck Framework. Die Yahoo! User Interface (YUI) Library ist ein Bündel an nützlichen Funktionen und Kontrollen in JavaScript für interaktive Webanwendungen. Es werden Techniken, wie DOM Scripting, DHTML und Ajax verwendet. Die YUI Bibliothek beinhaltet auch einige Kern CSS Ressourcen. Alle Komponenten der YUI Bibliothek werden als Open Source unter der BSC Lizenz vertrieben und sind somit für alle möglichen Zwecke bedenkenlos einsetzbar. Dieses Projekt startete 2005 und die aktuelle Version ist 2.5.0. Die Community ist wegen der bekannten Marke Yahoo groß, so haben sich in der Yahoo Gruppe mehr als 6.000 Mitglieder registriert und monatlich erhält der Blog mehr als 1.000 neue Einträge. Die Community wird stark durch das Yahoo Entwicklernetzwerk geprägt. Die Dokumentation besteht aus Online Manuals, Getting-Started Beispielen, einer API und einigen Entwickler Blogs. In der Gesamtheit betrachtet, handelt es sich um ein Framework, welches viele Eigenschaften von Desktop Applikationen unterstützt. Die verwendeten Technologien sind JavaScript, HTML und CSS.

YUI besteht aus zwei grundlegenden Typen, nämlich Utilities und Controls. Zu den Utilities zählt man Animationseffekte, Browser Chronik Management, Verbindungsmanagement (XMLHTTP Request Umschlag), Datenhaltungsmanagement für Benutzeroberflächen, DOM (HTML und CSS Kodierungsunterstützung), Drag und Drop und Ereignisse (inklusive Browser events). Zu den Controls zählen die herkömmlichen User Interface Elemente, wie z. B. eine Schaltfläche, ein Baum, ein Container, Reiter, Menüs, ein Slider etc. Das Aussehen kann mit CSS gestaltet werden. Die Architektur von YUI ähnelt der anderer JavaScript Frameworks. YUI ist ein mächtiges Werkzeug für den Client, muss jedoch auf der Serverseite zusätzlich ein Wrapper integriert werden:

---

<sup>130</sup> Vgl. Yahoo! UI (2007): „The Yahoo! User Interface Library (YUI)“

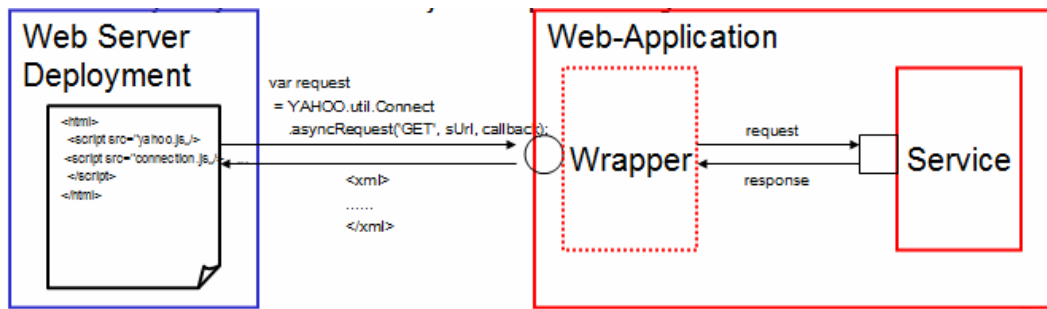


Abbildung 19: Yahoo Architektur

YUI ermöglicht die Kreation von Webapplikationen, die verblüffende Ähnlichkeit zu Desktop Applikationen aufweisen. Für ADOweb ist YUI eine exzellente Wahl bezüglich der Gestaltung der Benutzeroberfläche und RIA. Das zeigt die folgende Abbildung, die einen MS Word ähnlichen Texteditor, der komplexe Schriftstücke mit Hilfe vieler, verschiedener Formatierungen erstellen kann:

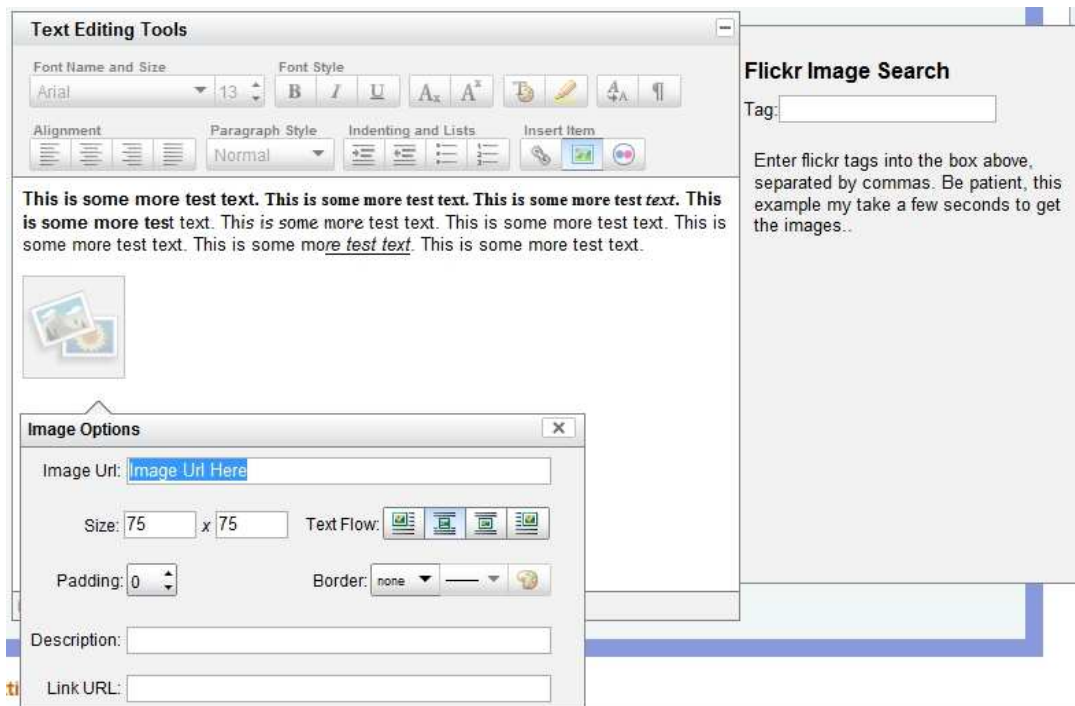


Abbildung 20: Yahoo! Text Editor

### 3.2.7 Rico



Open Rico<sup>131</sup> ist wie die meisten Ajax Frameworks clientseitig und mehrzweckfähig. Der Fokus des Frameworks liegt auf visuellen Effekten, die von einer Prototype Infrastruktur getragen werden. Rico wird seit 2005 entwickelt und die aktuelle Version ist 2.0. Die verwendeten Technologien sind HTML, JavaScript und Prototype. Für Rico müssen die Antworten von Webservices in einem speziellen Format aufbereitet werden, wie die folgende Architektur mit Hilfe eines Wrappers löst:

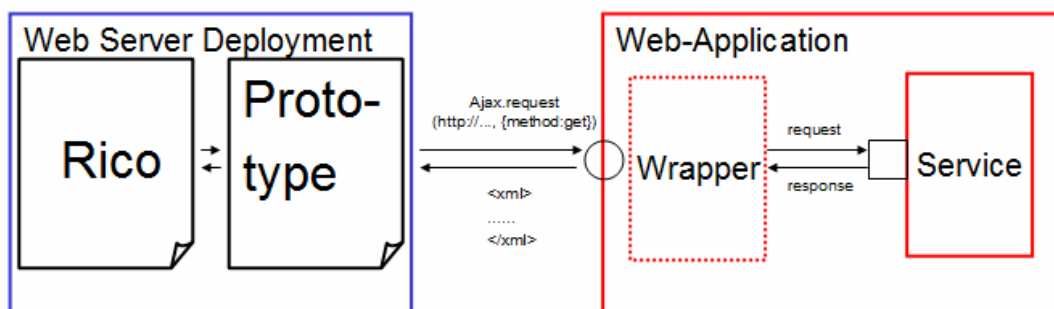


Abbildung 21: Rico Architektur

Die Rico Community umfasst 2.000 registrierte Benutzer und hat Referenzen zu Prototype. Die Qualität der Community ist schwer abzuschätzen, da das Forum für Support Anfragen gehandhabt wird. Weiters erhält Rico Unterstützung durch das Eclipse ATF Framework, welches das Editieren und Kompilieren des Codes erlaubt. Es werden einige gute visuelle Effekte und spektakuläre Widgets geliefert. Gewöhnliche Widgets müssen jedoch selbst implementiert werden. Das Framework ist aufgrund der Verbindung mit Prototype verlässlich und kann gut für die Produktion eingesetzt werden. Die Dokumentation hält sich in Grenzen und es gibt nur einige Beispiele, die eine gewisse Funktionalität vorweisen. Für die Erstellung von ADOweb RIAs ist dieses Framework eher unbrauchbar. Die Struktur einzelner Widgets, wie z. B. das in der nächsten Abbildung vorgestellte Wetter Widget, welches Fakten über die aktuellen Wetter- und Monddaten wiedergibt, sind für den Einsatz in ADOweb sinnvoll:

<sup>131</sup> Vgl. Fiveruns (2007): "Rico"



Abbildung 22: Rico Wetter

### 3.2.8 ASP.NET Ajax



Hinter dem Namen ASP.NET Ajax<sup>132</sup> lässt sich die Firma Microsoft vermuten, die schon mit dem ASP.NET Projekt eine Offensive auf das Internet gestartet hat. Es handelt sich um ein serverseitiges Framework, welches auf der Clientseite mit Hilfe von JavaScript Bibliotheken eine Verbindung herstellt. Die Bibliotheken beinhalten verschiedenste Ebenen, die die unterschiedlichen Browserabhängigkeiten abstrahieren können, um ein einheitliches Aussehen auf allen Browsern zu gewährleisten. Die Applikation wird von einem ASP.NET kompatiblen Server, wie z. B. IIS oder Abyss Web Server, gesteuert, die den geeigneten JavaScript Code anspricht. ASP.NET Ajax wird seit 2005 entwickelt und hat es bis heute zur Version 1.0 geschafft. In den vorherigen Versionen war es unter Atlas bekannt. Die Community wird von professionellen Microsoft Mitarbeitern betrieben. Die verwendeten Technologien sind HTML, JavaScript und ASP.NET 2.0. Die Clientseite wird in JavaScript oder XML Script Code geschrieben. Um die Verbindung zu einem laufenden Webservice herzustellen, ist es nötig, eine Bridge zu deklarieren. Dieser Vorgang umfasst das Schreiben einer „asbx“ Datei in XML Syntax und den Einsatz eines ASP.NET kompatiblen Server, wie z. B. IIS:

---

<sup>132</sup> Vgl. Microsoft (2007): „ASP.NET“

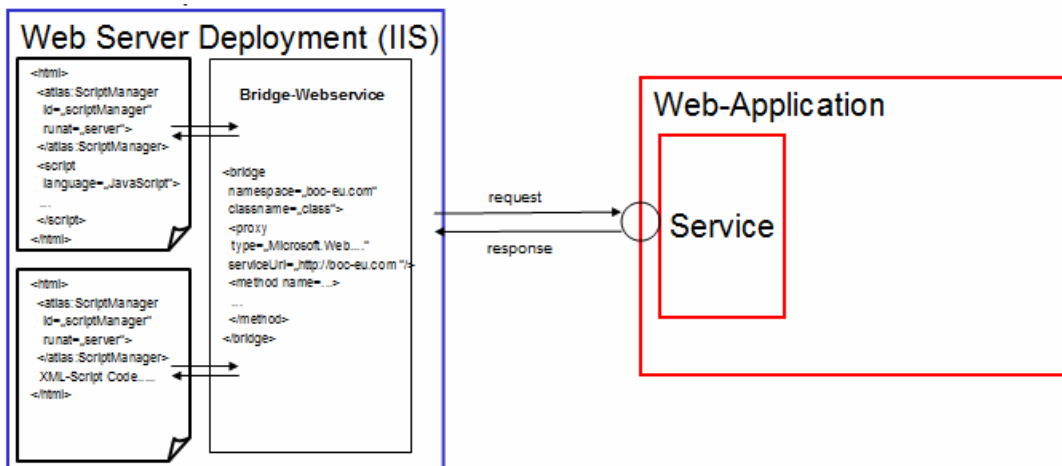


Abbildung 23: ASP.NET Architektur

Es werden JavaScript, HTML, XML Script, ASP.NET – Webservice Syntax (optional C# für eigene Webservices) verwendet. Wenn mit C# entwickelt wird, empfiehlt sich das Microsoft Visual Studio oder SharpDevelop. Die Community ist aufgrund der Bekanntheit von Microsoft ca. 300.000 registrierte Mitglieder, die pro Tag 2.000 Beiträge beisteuern. Microsoft bietet Unterstützung, wie z. B. freies Hosting. Die Dokumentation und die Ausreifung sind zufrieden stellend, obwohl während der Laufzeit Probleme auftreten. Browser Abhängigkeitsprobleme sind bei Safari bekannt, während Internet Explorer und Mozilla Firefox unterstützt werden. Für ADOweb RIAs ist dieses Framework eher nicht geeignet, da proprietäre Software für Entwicklung und Laufzeit verwendet muss. Webservices müssen in Brücken eingebunden werden und die spezifische Syntax von ASP.NET bei JavaScript und XML Skripten muss eingehalten werden. Dadurch wirkt es in der Handhabung recht unflexibel. Abgesehen davon ist die Benutzerflächengestaltung mächtig. Dies wird in der folgenden Abbildung einer Slideshow Demonstration gezeigt, die eines von vielen Controls bzw. Widgets (linke Spalte) von ASP.NET darstellen:

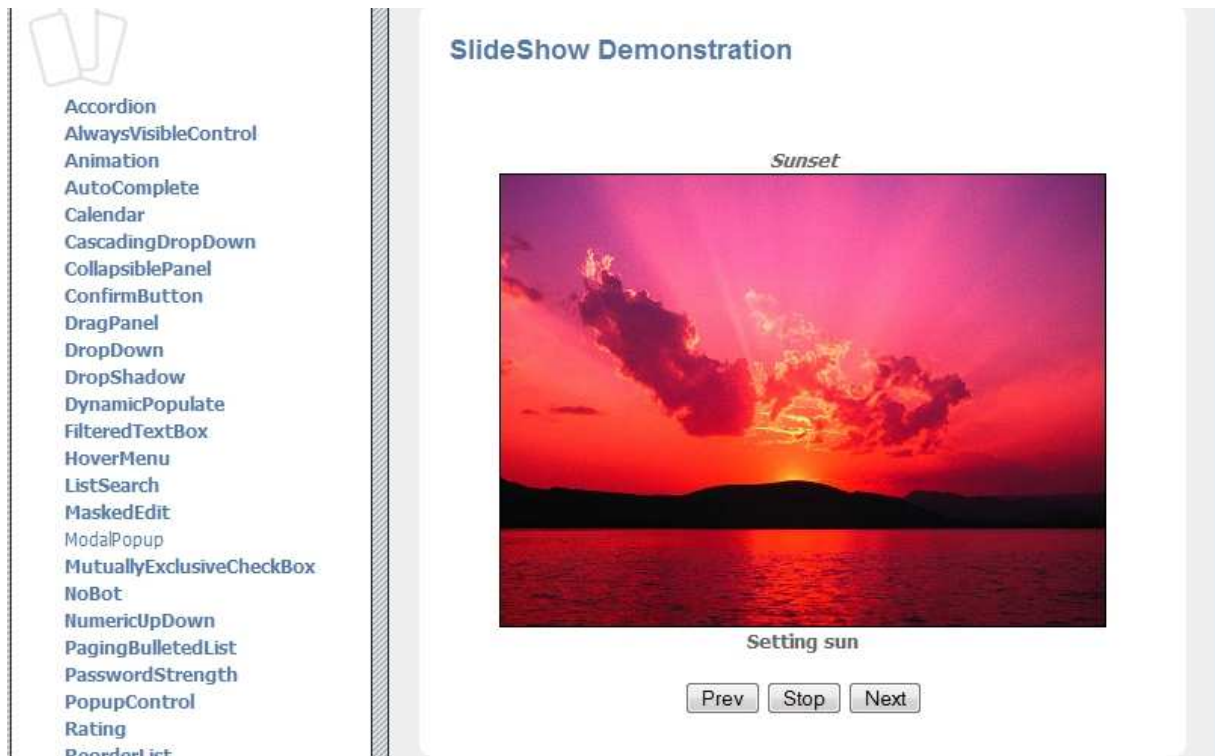


Abbildung 24: ASP.NET Slideshow und Toolkit

### 3.2.9 Xajax



Bei Xajax<sup>133</sup> handelt es sich um ein PHP basiertes serverseitiges Framework. Dafür wird eine PHP Klassenbibliothek bereitgestellt, die in der Lage ist, ohne die Hilfe von JavaScript Code den Inhalt von HTML Seiten zu gestalten. Dazu werden die benötigten JavaScript Funktionen in PHP Funktionen verpackt, die für die dynamische Generierung Sorge tragen. HTML, JavaScript und PHP sind die vorrangig verwendeten Technologien. Auf dem Web Server benötigt zur Ausführung von PHP Applikationen eine PHP Laufzeitumgebung. Der Zugriff auf eine externe Webapplikation kann mittels einer HTTP Anfrage realisiert werden:

<sup>133</sup> Vgl. White Jared und Wilson, Max (2007): „Xajax“



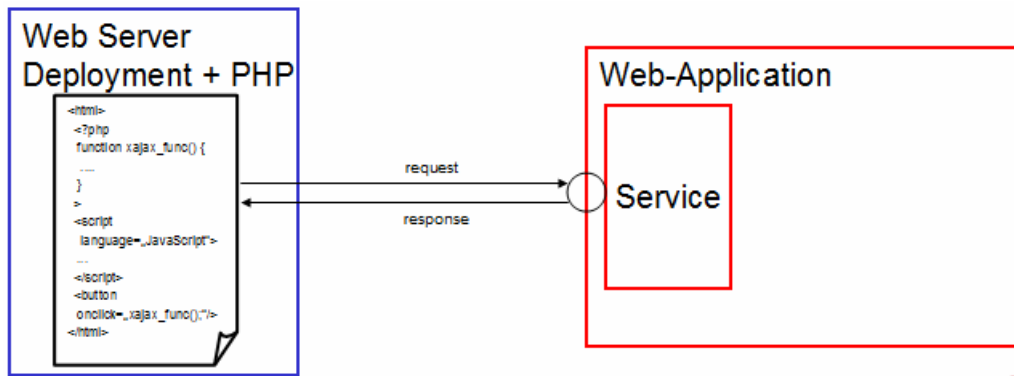


Abbildung 25: Xajax Architektur

Die Community umfasst ca. 2.000 Mitglieder. Sie besitzt Spezialwissen in PHP, welches von zahlreichen Foren und Wikis unterstützt wird. Es werden auch Plugins entwickelt, die das Xajax Framework erweitern. Der Reifegrad und die Dokumentation sind zufrieden stellend, jedoch wirken sich die begrenzten Möglichkeiten der serverseitigen Skriptsprache PHP zu Ungunsten einer komfortablen Applikationsoberfläche aus. Dieses Framework ist zwar gut für neue Projekte, jedoch ungeeignet für die Integration in das bestehende ADOweb Framework. Die GUI Funktionalität ist auch nicht beeindruckend. die in der folgenden Abbildung exemplarisch anhand einer Graphity Wall gezeigt wird, die mit Einträgen in verschiedenen Farben und Schriftgrößen „beschriftet“ werden kann:



Abbildung 26: Xajax Graphity Wall



### 3.2.10 GWT

Bei GWT<sup>134</sup> (Google Web Toolkit) von Google handelt es sich um ein java-basiertes serverseitiges Framework. Die Veröffentlichung der Version 1.0 erfolgte im Mai 2006 und ein Jahr später brachte man es zur Version 1.4 unter der Apache 2.0 Lizenz. Einige Applikationen dieses Frameworks, wie z. B. Gmail oder Google Maps oder Earth, erreichten schon große Popularität. Neben Google unterstützt auch die Firma JetBrein dieses Projekt. Das Forum hat mehr als 30.000 Einträge, die sich rund um GWT und die eingesetzten Technologien Java und HTML drehen.

Das Besondere am GWT ist, dass die Programmiersprache Java als Designkomponente verwendet wird. Die Java Klassen werden dann in eine Webapplikation, die auf HTML und JavaScript basiert, übersetzt. Zu den Eigenschaften des GWT zählen dynamische, wieder verwendbare UI Komponenten, Browser Historie Verwaltung und Kompatibilität, einfache RPC (Remote Procedure Call) und JSNI (JavaScript Native Interface). Die Daten werden durch einen eigenen Server zur Verfügung gestellt, der von einem betriebssystemabhängigen Batch File gestartet wird. Der Betrieb einer GWT Applikation auf einem anderen Server, wie z. B. dem weit verbreiteten Apache, ist nicht möglich. Zur Entwicklung bieten sich Java Entwicklungsumgebungen, wie z. B. Eclipse, wertvolle Unterstützung durch Typprüfung oder JavaScript Fehlererkennung zur Compilezeit an. Entwickler, die aus solchen Umgebungen kommen, werden das Erstellen von Widgets und dessen Ereignisverarbeitung mögen, da die API der von SWING und SWT sehr ähnelt. Die Transformation von Java in JavaScript benötigt nicht java-basierten Code in einer XML und HTML Dateien, um die Serverobjekte mit den Clientobjekten zu assoziieren. Die Applikation weist im Browser ein innovatives und eigentümliches Aussehen auf und besticht durch die Reichhaltigkeit und Effizienz der Widget Bibliothek.

Die Kommunikation mit externen Webservices erfolgt entweder über RPC oder das direkte Ansprechen der WSDL Schnittstelle mittels einer URL. Im Java Kontext bedeutet dies, dass Stubs generiert werden, die die Services mit Hilfe einer internen Logik aufrufen:

---

<sup>134</sup> Vgl. Google (2007): „GWT (Google Web Toolkit)“

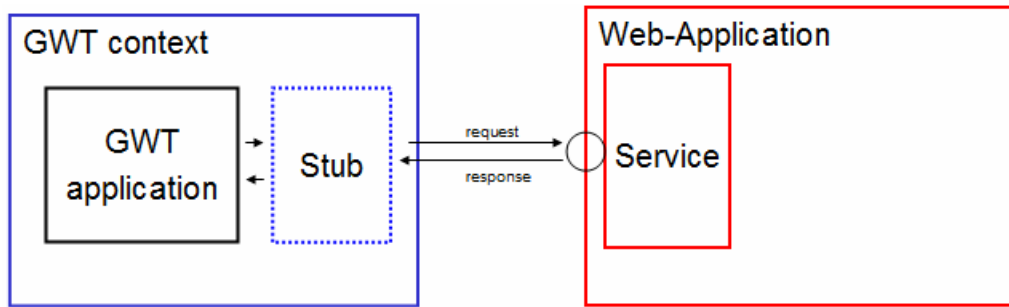


Abbildung 27: GWT Architektur

Für ADOweb ist dieses Framework aufgrund Einschränkungen in der kommerziellen Nutzungslizenz problematisch. Des Weiteren werden beim Starten des „Web Hosted Mode“ (Testmodus für GWT Applikationen) private Daten, wie z. B. die neueste Browserversion oder die IP-Adresse, an den Google Webserver verschickt. Google hat ein neues Projekt im Auge, welches auf der Java Swing API basiert und ähnlich zu RAP dynamisch JavaScript erzeugt. Es ist nicht auszuschließen, dass das GWT in naher Zukunft als obsolet zu betrachten ist. Dennoch ist das GWT im Rahmen seiner Möglichkeiten sehr mächtig. Die untenstehende Applikation zeigt einen webbasierten Email Client, der durch Effizienz und Komfort besticht:

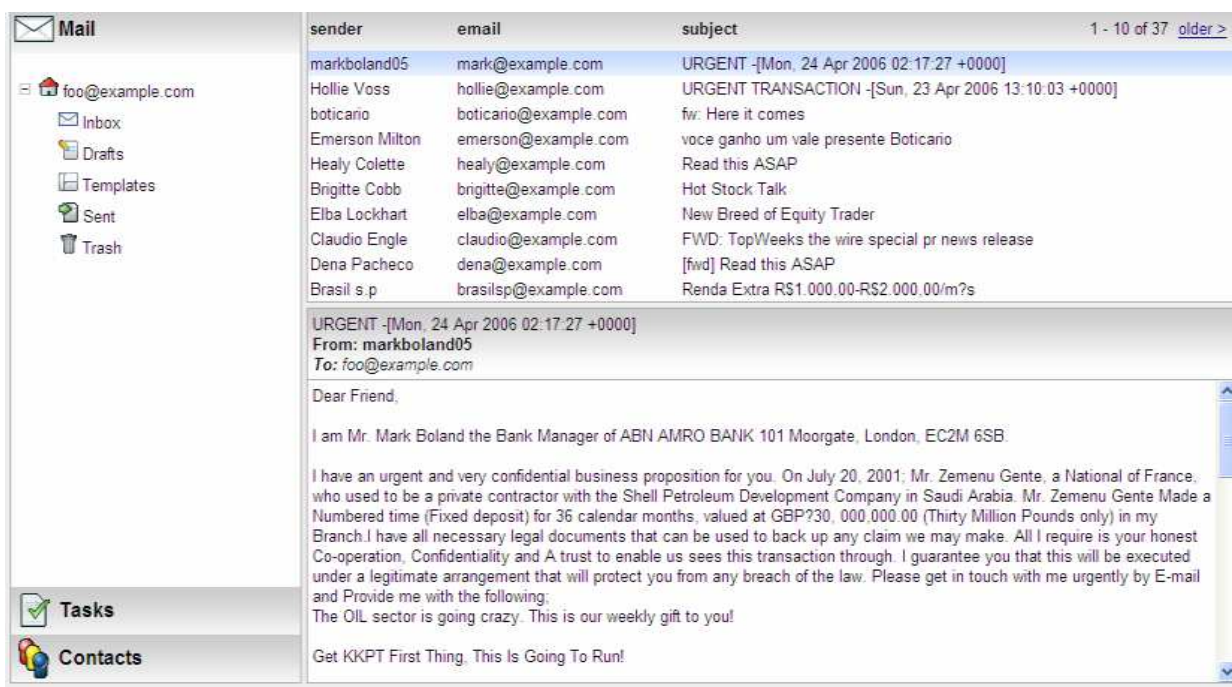


Abbildung 28: Google Mail

### 3.2.11 ThinWire



Bei ThinWire<sup>135</sup> handelt es sich um ein serverseitiges Java Framework. Die ganze Applikation wird wie eine Desktop Applikation in Java programmiert, wobei eine mitgelieferte Widget Bibliothek für maximalen Entwicklungskomfort sorgt. Dieses Projekt ist ein im Jahr 2003 gegründetes Spin-Off Unternehmen eines internen Kreditabrechnungssystems, und wird mittlerweile schon professionell in der Produktion eingesetzt. Die reine Java Webapplikation kann in einem Servlet Container eingesetzt werden. Eine Java Klasse muss dann nur mehr als Eingangspunkt (Entry point) der Applikation in einer Konfigurationsdatei deklariert werden. Das ThinWire System übersetzt die Java Klassen in äquivalente JavaScript und HTML Dateien, die an den Browser auf Anfrage geschickt werden. Die Natur eines verteilten Systems wird dem Entwickler durch die Transparenz des Java Codes vollkommen verborgen. Im Gegensatz zu GWT kann ThinWire zu hundert Prozent in Java programmiert werden, was bedeutet, dass die Entwicklung beinahe der einer Desktop Applikation gleicht. Zur Kommunikation mit einem dritten Service wird wie bei GWT ein Stub benötigt, der eine Schnittstelle bzw. Zugriffslogik zum Service hält:

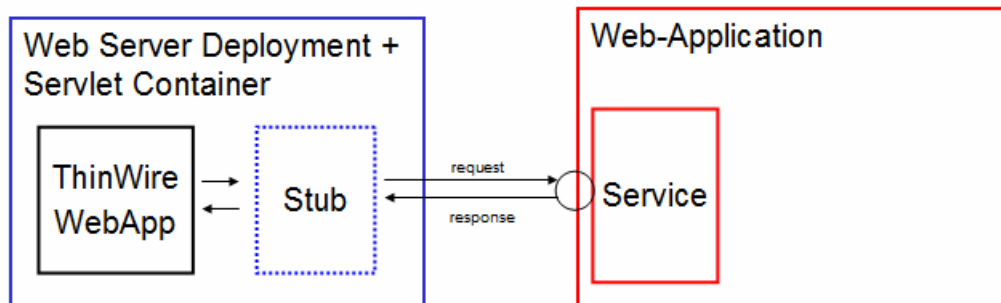


Abbildung 29: ThinWire Architektur

Der GUI Java Code ähnelt dem Swing Toolkit, ist aber aufgrund der verteilten Programmierung naturgemäß etwas abgewandelt, wie z. B. einem Upload Service. Mit Eclipse und einem zusätzlichen GUI Builder „Form Creator“ kann die Applikation komfortabel entwickelt werden. Es handelt sich hierbei um eine kommerzielle Lösung, die von Support und Feature Anfragen lebt (ab \$ 999 pro Jahr). Deshalb existiert hier keine offene Community. Kooperationen ergeben sich aus dem ursprünglichen Kundenkreditabrechnungssystemprojekt, welches das anfängliche Basis Framework für ThinWire bot. Mit ThinWire können leicht RIA Applikation erstellt und gewartet werden, da

<sup>135</sup> Vgl. ThinWire LLC (2007): „ThinWire“

viele Widgets und Event Handling unterstützt werden. Eine API ist verfügbar, jedoch ist der Support für neue Mitglieder kostenpflichtig. Darüber hinaus werden alle gängigen Browser unterstützt und es eignet sich hervorragend für die Umsetzung von RIAs. Außerdem eignet sich das Framework ausgezeichnet für die Zwecke von ADOweb und sollte für die Implementierung der Literaturdatenbank ernsthaft in Erwägung gezogen werden. Die GUI Funktionalität soll durch die folgende Abbildung eines CRM Systems gezeigt werden, dass Menüs, Bäume Reiter, Formulare etc. beinhaltet:

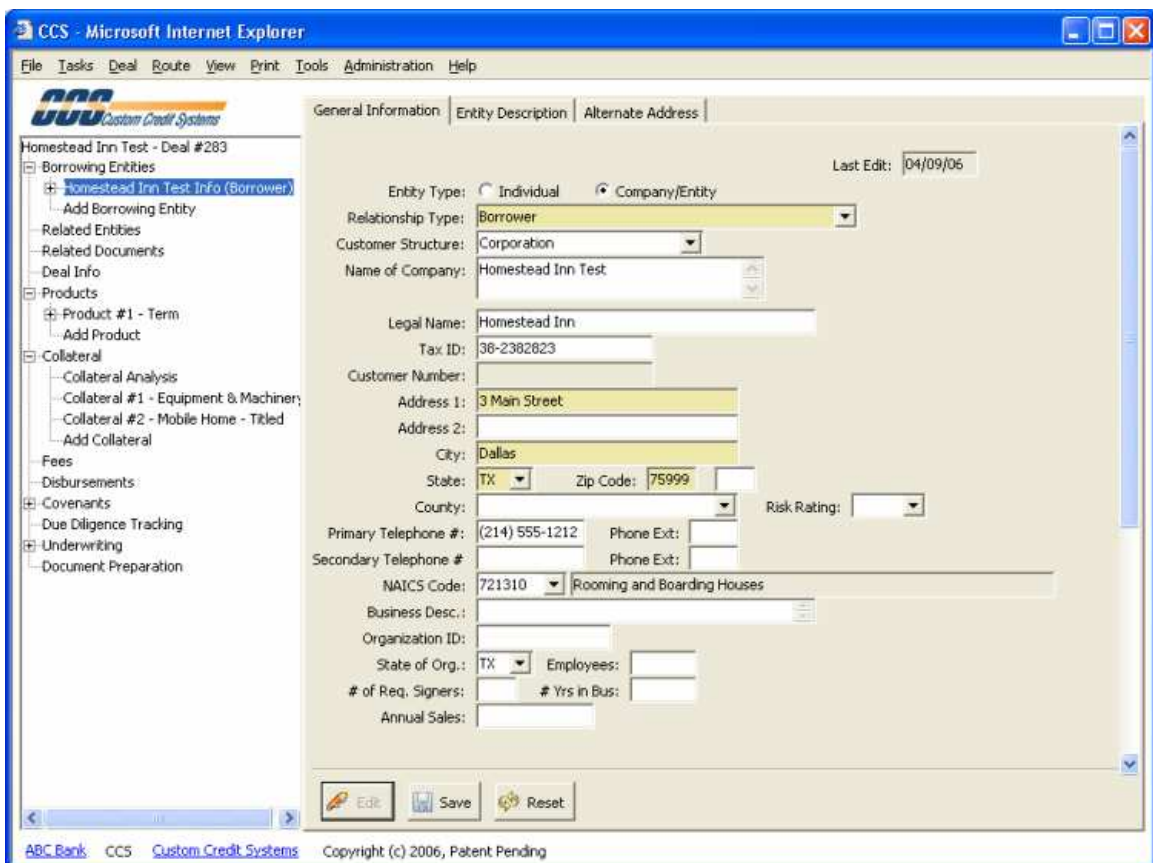
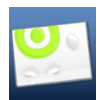


Abbildung 30: ThinWire CRM System

### 3.2.12 Open-jACOB



OpenJacob<sup>136</sup> ist ein Java Framework zur Erstellung von RAD (Rapid Application Development) Applikationen. Die aktuelle Version ist 2.7.2. Auf der Webseite steht ein Video zur Verfügung, welches verspricht, innerhalb weniger Minuten eine lauffähige Ajax Applikation erstellen zu können. Das Framework wird als Eclipse Plugin ausgeliefert, dass für

<sup>136</sup> Vgl. Herz, Andreas (2007): „Open-jACOB“

Entwickler eine Benutzeroberfläche bereitstellt, welches mittels Drag & Drop Aktionen Datenquellen an bestimmte Benutzeroberflächenelemente binden kann. Sogar Personen mit kaum bzw. gar keiner Programmiererfahrung können in einfacher Weise Webapplikationen erstellen. Die verwendeten Technologien sind Java, JSP, Struts, JavaScript, und HTML.

Der Komfort bei der Entwicklung ist auf die Anbindung an eine Datenbank zugeschnitten. Dies stellt eine Inkompatibilität mit ADOweb dar, welche Daten aus dynamischen Skripten bzw. aus Webserviceabfragen generiert. Deshalb wird hier auf eine Architektur verzichtet. Eine Eigenschaft von OpenJacob ist die JavaScript Bibliothek namens Draw2D, welches grafische Workflow mit Hilfe von Mausgestiken zeichnen kann, dass vor allem für das Modellieren in ADONIS relevant ist:

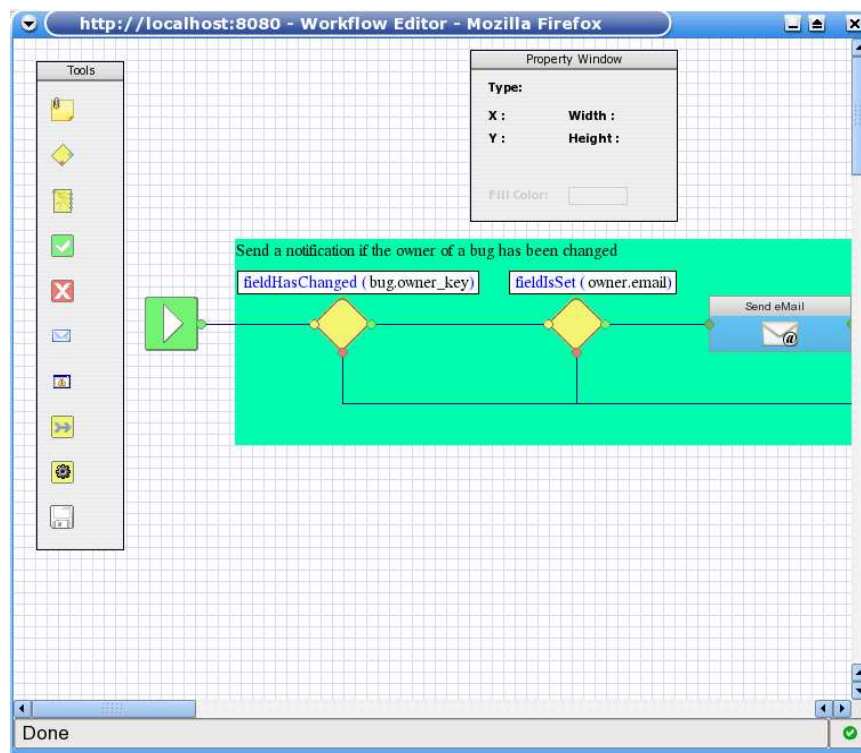


Abbildung 31: Draw2D von Open-jACOB

### 3.2.13 Qooxdoo



Qooxdoo<sup>137</sup> leitet sich aus dem Süddeutschen „Guckst Du“ ab und steht für ein clientseitiges Framework. Die JavaScript Bibliothek ist in einer Klassenhierarchie angelegt, die sämtliche Merkmale von objektorientierten Programmiersprachen, wie z. B. Vererbung, aufweist.

<sup>137</sup> Vgl. 1 & 1 Internet AG (2007): „Qooxdoo“

Qooxdoo kapselt die Client-Server Kommunikation und stellt darüber hinaus ein Widget Toolkit zur Verfügung, welches sich nicht mehr stark von herkömmlichen Desktop Applikation unterscheidet. Qooxdoo wird seit 2003 von 15 Entwicklern der 1 & 1 Internet AG vorangetrieben und hält bei der Beta Version 0.8 fest. Die Community kann sich über Neuigkeiten und Blogs informieren, wobei sich die Anzahl der Mitglieder schwer abschätzen lässt. Die verwendeten Technologien sind JavaScript und HTML. Ein HTML Dokument enthält alle nötigen Verweise zu Qooxdoo JavaScripten. Externe Services werden über einen Wrapper indirekt angesprochen, wie die nachstehende Grafik veranschaulicht:

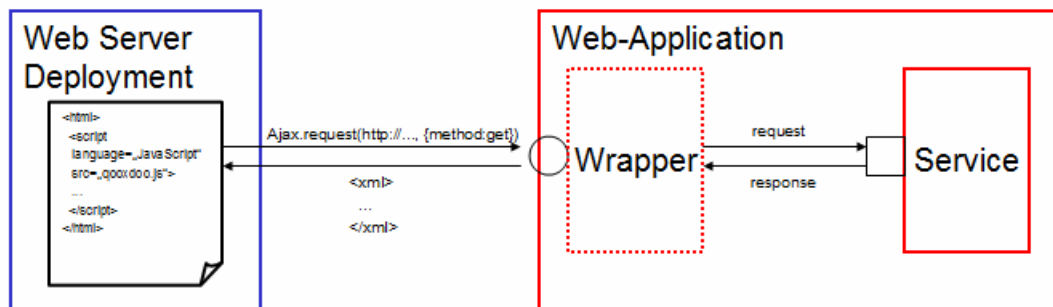


Abbildung 32: Qooxdoo Architektur

Die zur Verfügung gestellte API (Advanced Programm Interface) ist sehr mächtig. Es werden sehr viele Widgets und Ereignisse unterstützt, wie die nächste Abbildung zeigt:

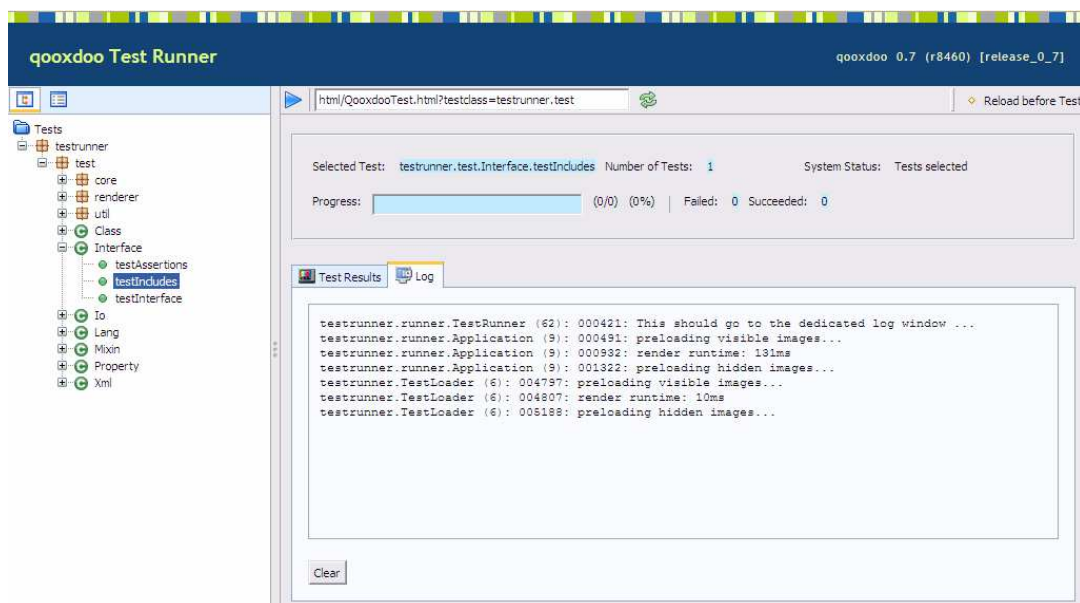


Abbildung 33: Qooxdoo Test Runner

Der GUI Code besteht ausschließlich aus JavaScript und HTML, wobei u. a. JSEclipse einen guten Editor bereitstellt. Die Eignung für eine Benutzeroberfläche von ADOweb ist aufgrund



Aussehen und Performanz exzellent. Qooxdoo wird sehr stark mit RAP in Verbindung gebracht, da RAP Qooxdoo als Rendering Komponente benutzt, wie der nächste Abschnitt schildert.

### 3.2.14 RAP



RAP<sup>138</sup> (Rich Ajax Application) ist eines der vielen Projekte, die von der Eclipse Community betrieben werden. Es gewann 2006 den ersten JAX Innovationspreis und befindet sich zurzeit in der Validierungsphase für ein Open Source Projekt. Im September 2007 wurde die Version 1.0 veröffentlicht. RAP stützt sich als RIA auf ein Bündel Eclipse Plugins, die über das Equinox/Osgi Framework als Applikationsserver gestartet werden. RAP benutzt zur Gestaltung der Benutzeroberfläche Java Bibliotheken, die ähnlich SWT und JFace sind. Hinzu kommt die mächtige Eclipse Entwicklungsumgebung, deren zahlreiche Funktionen, wie z. B. Debuggen oder Refactoring, eine äußerste komfortable Entwicklung ausschließlich in Java ermöglicht. Jochen Krause, der CEO von Innopract und Mitglied des Vorstands der Eclipse Foundation, behauptet, das mit Hilfe dieser drei Zutaten, also Widget-Toolkit, Plug-in-Mechanismus und eine Workbench, eine völlige Transparenz bei der Konstruktion von Ajax Applikationen gewährleistet werden kann. Dies bedeutet, dass nur der Ersteller eines Widgets die damit verbundenen Technologien beherrschen muss, und alle Entwickler von Anwendungen nur dessen Java API verwenden.<sup>139</sup> Um Ajax Applikationen zu ermöglichen, wird Qooxdoo als Rendering Komponente für den Browser verwendet. SWT bzw. JFace werden bei Laufzeit in äquivalente Qooxdoo JavaScript Klassen übersetzt, wobei während der gewöhnlichen Entwicklung keine einzige Zeile JavaScript oder HTML zu sehen ist. Der Programmierer hat das Gefühl eine RCP Desktop Applikation zu entwickeln, und tatsächlich kann durch Austauschen der Plugin Plattform die Applikation mit wenigen Adaptionen auf einem beliebigen Betriebssystem laufen.

Dieses relativ junge Projekt ist noch recht unbekannt, die Zielsetzung ist jedoch sehr ansprechend. Andere Projekte wählen einen ähnlichen Ansatz, der auf der SWING API beruht.<sup>140</sup> In der Newsgroup sind im letzten Jahr etwa 400 Beiträge eingegangen. Des

---

<sup>138</sup> Eclipse Foundation (2007): „RAP – Rich Ajax Platform“

<sup>139</sup> Meyen, Sebastian (2006): „EclipseCon: Eclipse Rich Ajax Platform - ein Interview mit Jochen Krause“

<sup>140</sup> CreamTec (2007): „AjaxSwing“

Weiteren ist die Kompatibilität mit anderen Systemen bemerkenswert, da eine RAP Applikation nicht nur auf einem Equinox Osgi Server, sondern auch auf anderen Servlet Containern, wie z. B. Jetty und Tomcat, laufen kann. Die nächste Abbildung zeigt eine solche Architektur einer RAP Applikation mit Einbindung eines dritten Service:

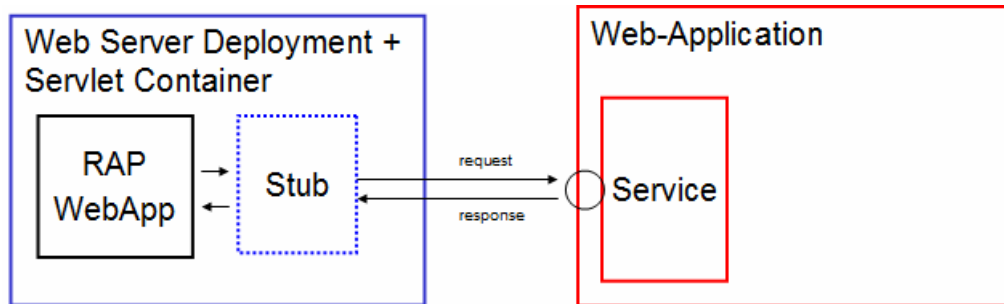


Abbildung 34: RAP Architektur

Die Dokumentation ist etwas dürftig und mehr Beispiele und Manuals wären wünschenswert. Für RIAs eignet sich dieses Framework aufgrund der reichhaltigen Benutzeroberflächenbibliotheken und die Portabilität auf Betriebssysteme trotz seiner Unreife hervorragend. Dieses Framework wird für die Zusammenarbeit mit ADOweb wärmstens empfohlen. Die Benutzeroberfläche ist beeindruckend und besteht aus einer Workbench, die einer herkömmlichen RCP (Rich Client Platform) zum Verwecheln ähnlich sieht:

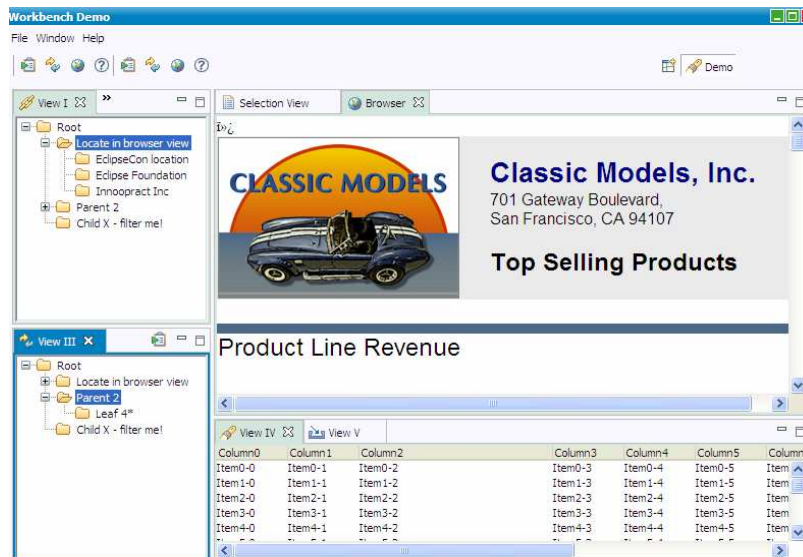


Abbildung 35: RAP Workbench



### 3.3 Ergebnisse der Analyse

Im den vorherigen Abschnitten wurden 14 Frameworks vorgestellt, die nach einer ersten Entscheidung als die wichtigsten eingestuft wurden. Ajax erfährt zurzeit einen rapiden und dynamischen Fortschritt und die Ergebnisse der Recherche sollten lediglich als temporäre Aufnahme des Entwicklungsstandes gesehen werden. Die folgende Analyse erfolgt mittels eigens definierter Selektionsparameter, an denen die Frameworks gemessen werden sollen. Die Parameter ergeben sich einerseits aus dem subjektiven Vorwissen des Autors, wie z. B. Kenntnisse und Erfahrung mit der Programmiersprache Java, und andererseits sollen objektive Parameter gemessen werden. Dabei ist der Ajax Schlüsselaspekt der Kommunikation wichtig, sowie die reichhaltige Applikationen zu entwickeln. Des Weiteren ist auch die Unterstützung des Browsers, sowie die Ausreifung des Produkts essentiell:

- *Kommunikation* (Abstraktion der asynchronen Kommunikation): Das Frameworks bietet „Ajax-Objekte“ für eine Verbindung zwischen Browser und Server. Eine gemeinsame Schnittstelle „XmlHttpRequest“ sorgt für die Reduzierung der Komplexität der Client-Server Kommunikation.
- *Applikationsentwicklung*: Dieser Parameter misst die Fähigkeit, grafische Benutzeroberflächen zu entwickeln, die die Browser Abhängigkeiten überwinden. Dazu benötigt das Framework ein “View and Control” Prinzip, meist in Form von Widgets und Controls.
- *Java Server*: Dies bezeichnet die Fähigkeit, dass das Framework mit einem serverseitigen Java Komponente, wie z. B. dem ADOweb Framework, kommunizieren kann.
- *Pure Java*: Dies bezeichnet die Eigenschaft, ob die gesamte Ajax Applikation in Java geschrieben werden kann.
- *Reife*: Dies beschreibt den Reifegrad eines Frameworks, d. h. ob es schon in der Produktion/Entwicklung eingesetzt werden kann oder ob es sich um eine Beta-Version handelt.
- *Browser Unterstützung*: Dies bezeichnet die Fähigkeit des Frameworks auf verschiedenen Webbrowsern zu laufen. In diesem Zusammenhang werden nur die am weitesten verbreiteten Browser in Betracht gezogen, wie Internet Explorer, Mozilla Derivate, Safari und eventuell Opera.

Die nächste Tabelle gibt einen Überblick über alle Kriterien der vorgestellten Frameworks:

	<i>Kommunikation</i>	<i>Applikationsentwicklung</i>	<i>Java Server</i>	<i>Pure Java</i>	<i>Reife*</i>	<i>Browser** Unterstützung</i>
Prototype	eingebaut	nur HTML			Produktion	<b>IE, F, S, O</b>
Script.aculo.us	Prototype	nur Effekte			Produktion	<b>IE, F, S</b>
Dojo	Prototype	gut			Beta	<b>IE, F, S, O</b>
DWR	eingebaut	nur HTML	X		reif	<b>IE, F, S, O</b>
jQuery	eingebaut	gut			reif	<b>IE, F, S, O</b>
Yahoo! UI	eingebaut	Sehr gut			reif	<b>IE, F, S, O</b>
Rico	Prototype	ausreichend			reif	<b>IE, F, S, O</b>
ASP.NET	eingebaut	gut			reif	<b>IE, F, O</b>
Xajax	eingebaut	nur HTML			reif	<b>IE, F</b>
GWT	eingebaut	Sehr gut	X		reif	<b>IE, F, S, O</b>
ThinWire	eingebaut	Sehr gut	X	X	Produktion	<b>IE, F, S, O</b>
OpenJacob	eingebaut	ausreichend	X	X	reif	<b>IE, F</b>
Qooxdoo	eingebaut	Sehr gut			Beta	<b>IE, F</b>
RAP	Qooxdoo	Sehr gut	X	X	Validierung	<b>IE, F</b>

\*

\*\*

Produktion	= professionelle Entwicklung	IE = Internet Explorer	>= 6
Reif	= stabile Version (>= 1.0)	M = Firefox	>= 1
Beta	= frühe Version (< 1.0)	S = Safari	>= 2
Validierung	= initiale Version	O = Opera	>= 8

**Tabelle 4: Ajax Framework Vergleich**

Alle Frameworks besitzen eine Kommunikationskomponente, die Anfragen und Antworten über das HTTP Protokoll mit einem Server erlaubt. Diese wird entweder als fester Bestandteil mitgeliefert oder mit Hilfe eines anderen Frameworks erfüllt, wie z. B. Prototype oder Qooxdoo. Die Bewertung der Applikationsentwicklung hängt von der Reichhaltigkeit und Aussehen der Widget Bibliothek und dessen Ereignisverarbeitung ab. Wenn ein Framework keine Widget Bibliothek besitzt, wird es mit der Bezeichnung „nur HTML“ versehen. Bei Vorhandensein wird sie mit einer Note charakterisiert oder im Falle von Script.aculo.us als Effektframework bezeichnet. „Java Server“ beschreibt die Einbindung der Programmiersprache Java auf der Serverseite und „Pure Java“ die ausschließliche Entwicklung damit. Die Parameter Reife und Browser Unterstützung werden in der Legende unter der Tabelle erklärt.

In der obigen Auswertung werden die Frameworks als Stand-Alone Lösungen betrachtet. Durch die Kombination mit anderen Frameworks kommen meist erst die Stärken der einzelnen Frameworks zu Tage, die gewisse Funktionalitäten besser erfüllen können. Die nächste Abbildung zeigt Vorschläge, wie die vorgestellten Frameworks miteinander gekoppelt werden können. Die Aufgabenteilung erfolgt in Kommunikation, Client Applikation und Server Framework. Jedes Framework kann für eines dieser Bereiche die Verantwortung übernehmen. Ein Mischen bzw. Überlappen ist meist aufgrund von Inkompatibilitäten der Frameworks nicht möglich. Zusätzlich erfolgt in der Spalte noch eine Angabe rechts außen über die Implementierung in der Programmiersprache Java, die aufgrund umfangreicher Vorkenntnisse bzw. Vorarbeit in ADOweb anderen vorgezogen wird.

Die folgende Tabelle zeigt die Auswertung, in der vier Frameworks bzw. Kombinationen von Frameworks empfohlen werden, die sich für eine komplette Webapplikationsentwicklung eignen. Nicht java-basierte Frameworks werden aufgrund des Nichtvorhandenseins überragender Eigenschaften automatisch ausgeschieden:

<i>Kommunikation</i>	<i>Client Applikation</i>	<i>Server Framework</i>	<i>Java</i>
Prototype			
Prototype	Script.aculo.us		
Dojo/Prototype	Dojo		
DWR	Dojo	DWR	
jQuery	jQuery UI		
Yahoo! UI	Yahoo! UI		
Prototype	Rico		
ASP.NET Ajax	ASP.NET Ajax	ASP.NET Ajax	
Open-jACOB	Open-jACOB	Open-jACOB	
xajax	xajax	xajax	
GWT	GWT	GWT	
ThinWire	ThinWire	ThinWire	
Open-jACOB	Open-jACOB	Open-jACOB	
Qooxdoo	Qooxdoo		
Qooxdoo	Qooxdoo	RAP	

Abbildung 36: Kombination und Auswahl von Frameworks

Einige clientseitige Frameworks enthalten dennoch interessante Funktionalitäten, die als wertvolle Bereicherung eingesetzt werden können und hier erwähnt werden sollen:

- *Prototype*: Es ist ein reifes Framework, welches die Ajax Kommunikation unterstützt. Prototype kann gut in existierenden Code eingebaut werden. Als Ergänzung kann es für clientseitiges Datenmodelle aus Performanzgründen verwendet werden.
- *Script.aculo.us*: Dies ist wiederum ein reifes und weit verbreitetes Framework. Es ist eine gute Ergänzung zu Prototype zur Unterstützung von visuellen Effekte und Drag & Drop.
- *Dojo*: Dojo hat einen Satz an Widgets, welches eine gute Ergänzung zu Prototype, ist. Außerdem kann Dojo SVG Grafiken interpretieren, was in Bezug auf grafische Modellierung sehr hilfreich sein kann.
- *OpenJacob*: Die Draw2D JavaScript Bibliothek empfiehlt sich sehr für die grafische Modellierung. Das eigentliche Framework ist aufgrund der Bindung an eine bestimmte Datenbank für ADOweb leider unbrauchbar.

Bei den vier Kandidaten ergibt sich folgendes Ranking:

- (1) *RAP*: RAP befindet sich noch in der Validierungsphase, wobei eine stabile Version Anfang Oktober 2007 erhältlich ist. Dieses Framework zeichnet sich durch Desktop Darstellung und Anwendbarkeit aus, und basiert vollständig auf der SWT (Standard Widget Toolkit) und JFace API. RAP wird von der bedeutungsvollen Community Eclipse gefördert. Dieses Framework empfiehlt sich ausgezeichnet für die Entwicklung von ADOweb, weil es unter der EPL (Eclipse Public License) frei erhältlich ist, und sofort eingesetzt werden kann.
- (2) *ThinWire*: Da eine stabile Version erhältlich ist kann diese sofort zur Entwicklung einer vollen Funktionalität eingesetzt werden. Das Aussehen der Webapplikation unterscheidet sich kaum mehr vom Desktop und die Entwicklung basiert auf einer Java SWING ähnlichen API. Die Unterstützung seitens des Unternehmens ist kostenpflichtig. Dieses Framework wird für sofortigen, professionellen Einsatz wärmstens empfohlen.
- (3) *DWR/Dojo*: Hinter DWR steht eine reife Community Unterstützung, bietet jedoch keine spezifische Widget Bibliothek. Daher wird Dojo als Front End Komponente empfohlen, die mittels eigener Brücken verbunden werden (glue-code). Die

Empfehlung ist mittelmäßig, da der Aufwand im Vergleich zu den vorher genannten Frameworks recht umfangreich sein wird.

- (4) *GWT*: Für *GWT* spricht ein reifer und starker Community Support. Das Aussehen richtet sich speziell auf Google spezifisches Look & Feel aus, benötigt aber eine eigene Laufzeitumgebung (Batch Dateien und Kompilierung). Außerdem ist ein neues Projekt in Aussicht, welches das *GWT* bald in den Schatten stellen könnte. Die Empfehlung ist mittelmäßig bis wenig.

Das Framework *ThinWire* hat eine mächtigere Bibliothek als *RAP*. Der Nachteil von *ThinWire* ist, dass es nicht Open Source und kostenpflichtig ist. Des Weiteren weist es mit Eclipse eine große und kompetente Community auf. Die anderen beiden Frameworks benötigen mehr Entwicklungsaufwand und Einarbeitungszeit und werden deswegen nur als Ersatz in Betracht gezogen. Die Entscheidung fällt deswegen zugunsten von *RAP* aus. Das nächste Kapitel schildert nun im Detail, wie der Prototyp auf diesem Framework aufgesetzt wird.

## 4 Die Implementierung

Dieses Kapitel befasst sich mit der praktischen Umsetzung einer Ajax Literaturdatenbank. Dazu wird das in Kapitel 4 ausgewählte Ajax Framework RAP verwendet, welches für die Gestaltung der Benutzeroberfläche auf Basis der Selektionsparameter in Abschnitt 3.3 ausgewählt wurde. Der eigentliche Inhalt der Literaturdatenbank wird mit der Wissensmanagementmethode PROMOTE modelliert. Die Verbindung des Modellbestandes mit der Weboberfläche erfolgt mittels ADOweb, welches über den ADONIS Webservice (AdoScriptService) Lese- und Schreiboperationen ausführt. Die Modelldaten werden für die Benutzeroberfläche mit einem speziellen Datenmodell (Content Provider der JFace Bibliothek) aufbereitet. Die Befüllung und die Änderungsüberwachung dieser Datenmodelle wird vom darstellungs- und datenherkunftsunabhängigen Cache der AdoScript Objects Bibliothek bewerkstelligt, die darüber hinaus das Verhalten der AdoScript Funktionen in Java Objekte kapselt. Zur Gestaltung der Benutzeroberfläche verwendet RAP die RWT Bibliothek, die eine Untermenge von SWT und JFace repräsentiert. Anstelle von binären Klassen generiert RWT dynamische Qooxdoo JavaScript Objekte und Methoden. SWT und RWT haben die gleiche Syntax, wodurch ein Auswechseln der beiden Bibliotheken denkbar ist. Der geschriebene RAP Webapplikationscode kann in einem Ausmaß von 80 bis 90% für eine RCP Stand-Alone Applikation wieder verwendet werden.

Der nächste Abschnitt klärt die Zielsetzung, die die grundlegenden Absichten dieser modellunterstützten Literaturdatenbank herausarbeitet. Dann erfolgt eine konzeptionelle Erläuterung der Architektur, welche die entwickelten Services in den Lebenszyklus von ADOweb Service Framework Komponenten einstuft. Danach wird der eigentliche Prototyp anhand einiger Screenshots beschrieben. Im Anschluss werden die aufgetretenen Probleme, sowie Einschränkungen besprochen, die Hindernisse bei der Umsetzung der Wunschvorstellungen darstellten. Einige Komplikationen konnten dank der RAP Community erfolgreich bewältigt werden. Durch aktive Teilnahme in der Newsgroup wurde es möglich einen Mausereignismelder (Mouse Listener) zu entwickeln, welcher einerseits die Plattform erweitert und andererseits interessante Einblicke in das Zusammenspiel der RAP Bestandteile erlaubt.

## 4.1 Zielsetzung

Die Wissensmanagementmethode PROMOTE besitzt die Fähigkeit Informationsräume abzubilden. Damit soll firmenrelevantes Wissen, welches für verschiedene Projekte verwendet wird, schnell und einfach verfügbar sein. Am konkreten Beispiel der EU-Gruppe der BOC Asset Management sind vier Datenräume von besonderem Interesse:

- Literaturdatenbank (physische Dokumente)
- ADOweb Repository (Entwicklung von ADOweb Webservices)
- Content Repository (elektronische Dokumente)
- Deliverables Repository (Projektangaben)

Bei der Literaturdatenbank handelt es sich um eine strukturierte Ablage von physischen Dokumenten, wie z. B. Magazine, Artikel, Workshop Material, Informationszettel oder Präsentationsfolien. Diese werden in der nächsten Abbildung als Wissensressourcen in der Ausprägung von Dokumenten und Wissensquellen beschrieben (siehe Block links außen):

### Akquisition, Annotation und Suche in der Literaturdatenbank

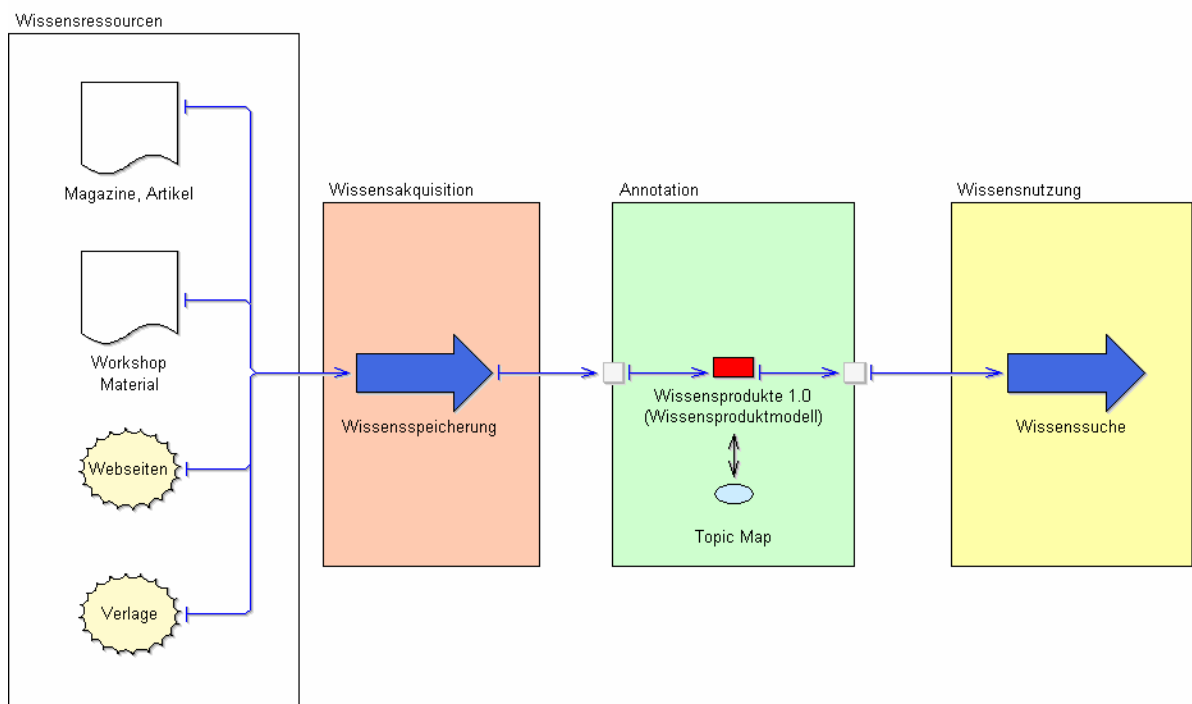


Abbildung 37: Anwendungsprozess der Literaturdatenbank

Die Metainformationen der realen Entitäten werden mit einfachen *Akquisitionsservices* in den Modellbestand importiert. Diese Beschreibungen werden einem oder mehreren Themen einer selbst definierten Topic Map zugewiesen (*Annotation*). Die Kategorisierung liefert den *Nutzungsprozessen* Wissen über den eigentlichen Inhalt der Wissensressourcen (siehe Block rechts außen). Zur Abwicklung eines solchen modellunterstützten Informationssystems werden drei Services zur Verfügung gestellt:

- Suchmaschine (Nutzung)
- Modell Editor (Akquisition & Modellierung)
- Modell Annotation (Annotation )

Der Hauptzweck der Literaturdatenbank besteht in der Bildung von Kompetenzen für Mitarbeiter eines Unternehmens, die mit den folgenden Maßnahmen gefördert werden sollen:

- Teilen und Wieder verwenden von Wissen
- Weiterbildung für Mitarbeiter
- Ermöglichen von produktiver Teamarbeit und gezielter Kommunikation
- Information auf Anfrage bereitstellen
- Vereinfachung des Informationszugriffs für (neue) Mitarbeiter

Die Aufgabenstellung soll jedoch nicht durch eine große, komplexe Stand-Alone Applikation, wie z. B. Lotus Notes, gelöst werden. Vielmehr soll die Funktionalität in einige, kleine Services aufgeteilt werden, die in Summe die Mächtigkeit des Systems ausmachen. Die folgenden Web 2.0 Konzepte sind von besonderem Interesse:<sup>141</sup>

- Services anstelle von Stand-Alone Applikationen
- Jeder kann teilnehmen (einstellen, lesen ,suchen)
- Nutzung der kollektiven Intelligenz durch Tagging
- Intuitive Ajax Benutzeroberfläche (Rich User Experience)

Wie diese Prinzipien in einem Prototyp umgesetzt werden, erläutert das nächste Kapitel.

---

<sup>141</sup> Vgl. O'Reilly, Tim (2005): „What is Web 2.0“





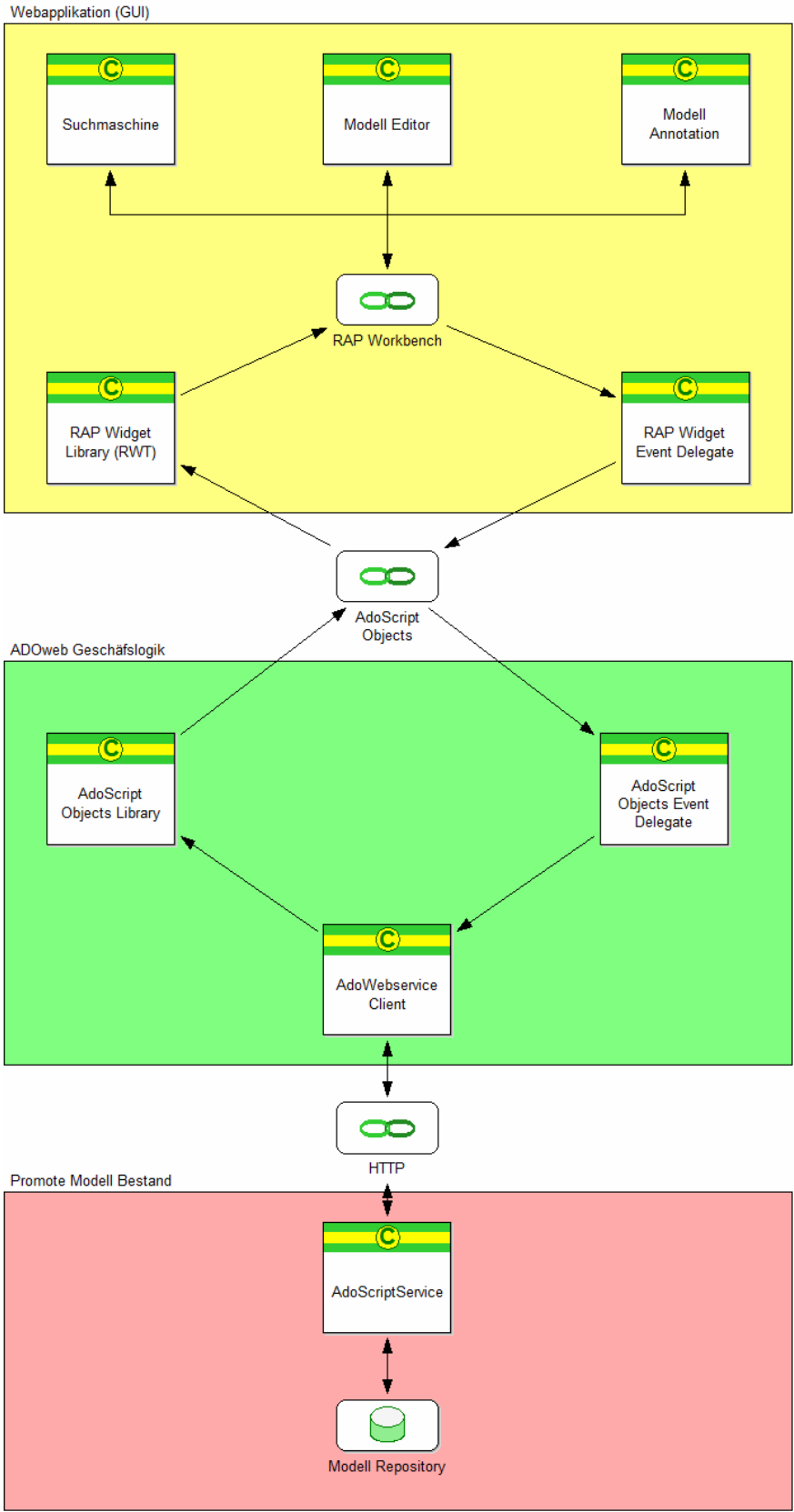
## 4.2.1 Software Bibliotheken

Die Implementierung verwendet Software Bibliotheken, die grundlegende Funktionalitäten bereitstellen, um die Entwicklungsarbeit zu erleichtern. Diese werden zur besseren Applikationsentwicklung bzw. -wartung mit dem MVC (Model-View-Control) Pattern strukturiert:

- PROMOTE Modell Bestand (Model)
- ADOweb Geschäftslogik (Control)
- Webapplikation (View)

Der PROMOTE Modellbestand wird mit Hilfe des ADONIS Webservice (*AdoScriptService*) über http verfügbar gemacht. Der *AdoWebservice Client* kann nun AdoScript Anfragen an den *AdoScriptService* schicken, um beliebige Bibliotheks-, Modell-, Objekt- bzw. Attributstatusinformationen abzufragen. Die Parameter der Anfragen sind meist datenbankspezifische Zahlen IDs (Integer) und die Antworten bestehen aus einfachen Zeichenketten oder XML Fragmenten. Zur Darstellung der Daten in der Benutzeroberfläche werden die AdoScript Methoden zur Gänze in Java Objekte (*AdoScript Objects Library*) gekapselt. Die Einschubung dieses Datenmodells kann die Daten in Objektform für eine beliebige GUI Technologie, sei es Ajax, JSP, JavaScript oder HTML, datenherkunftsneutral aufbereiten. Änderungen am *AdoScript Object* Datenmodell, wie z. B. das Umbenennen des Modellnamens oder das Setzen eines Objektattributs, werden durch den *AdoScript Objects Event Delegate* bewerkstelligt, der die notwendigen AdoScript Methoden zum Aktualisieren des Modellbestandes ausführt. Die Benutzeroberflächenelemente der *RAP Widget Library* (*RWT*) müssen dann nur mehr mit den Werten der AdoScript Objects befüllt werden, womit die Benutzeroberflächenentwicklung gänzlich von der Last der Datenhaltung befreit wird. Die einzelnen Widgets, wie z. B. Textfelder, Auswahllisten oder Bäume, werden ihrerseits vom *RAP Widget Event Delegate* überwacht, welches Ereignisse, wie z. B. die Eingabe eines Werts in ein Textfeld, an das AdoScript Objekt weiterleitet, das wiederum den *AdoScript Objects Event Delegate* anstößt. Mehrere Widgets werden in Views (Fenster) zusammengefasst und mehrere Views ergeben eine Perspective (Ansicht), die nichts anderes als die Services Suchmaschine, Modell Editor und Annotation der RAP Workbench sind. Ein gesamtes Bild des Zusammenspiels der erklärten Software Bibliotheken wird mit der nachstehenden Abbildung veranschaulicht:

**Literaturdatenbank Architektur**



**Abbildung 39: Software Bibliotheken der Ajax Literaturdatenbank**

Der nächste Abschnitt führt das Modell Repository, welches die Daten verwaltet, näher aus.

## 4.2.2 Modell Repository

Die Daten der Literaturdatenbank werden in PROMOTE modelliert, wobei zur Laufzeit drei Objekt- bzw. zwei Modelltypen abgefragt und manipuliert werden:

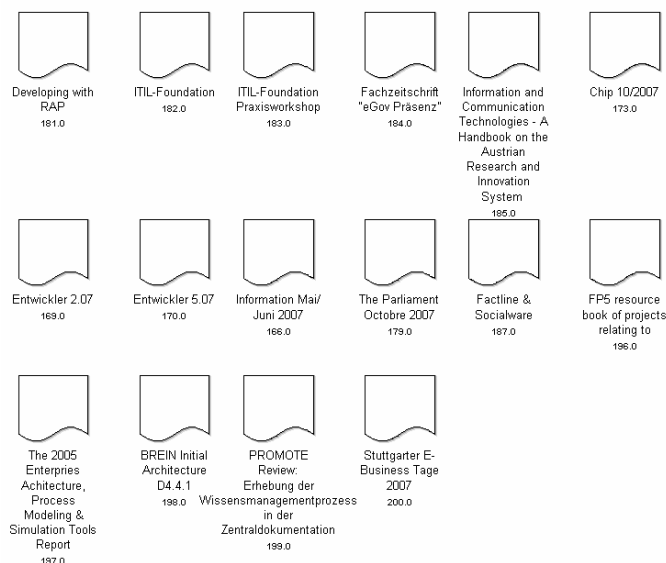
<i>Objekt</i>	<i>Modell</i>	<i>Zweck</i>
Dokument	Wissensressourcenmodell	Metadaten der Ressourcen
Wissensquelle	Wissensressourcenmodell	Herkunft der Ressourcen
Topic	Wissensstrukturmodell	Themenzuordnung

**Tabelle 5: Verwendete PROMOTE Modelltypen**

Das Objekt *Dokument* repräsentiert Metadaten, die den Inhalt der Ressource beschreiben. Diese werden mit den folgenden Attributen ausgestattet:

- *Name*: Titel der Dokuments, der Zeitschrift oder des Artikels
- *Beschreibung*: Kurze Erläuterung wichtiger Themen bzw. Seiten
- *Kommentar*: Schlagwörter, die sich auf dieses Dokument beziehen sollen
- *Identifizier*: Eindeutige Kennzeichnung, wie z. B. eine fortlaufende Nummer
- *Archiv*: Container bzw. Ort an dem sich das Dokument befindet
- *Besitzer*: Person, der dieses Dokument gehört

In PROMOTE werden die Dokumente in einem Wissensressourcenmodell abgelegt:

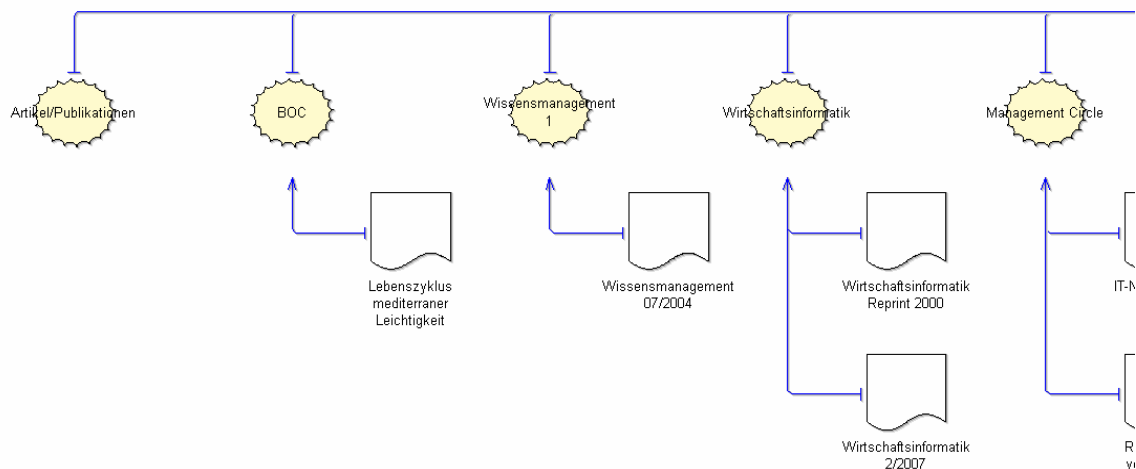


**Abbildung 40: Lose Dokumente in einem Wissensressourcen Pool**

Das Objekt *Wissensquelle* kennzeichnet den Ursprung des Dokuments, wie z. B. eine Webseite oder einen Verlag. Die interessanten Attribute bei diesem Objekt sind:

- *Name*: Bezeichnung der Herkunft des Dokuments
- *Beschreibung*: Kurze Erläuterung der Wissensquelle
- *Kommentar*: Schlagwörter, die sich auf diese Wissensquelle beziehen sollen
- *Link*: Referenz bzw. Hyperlink auf die Wissensquelle
- *Kosten*: Sagt aus, ob die Wissensquelle offen oder kostenpflichtig ist
- *Zugriffsart*: Sagt aus, ob der Zugriff kontinuierlich oder periodisch erfolgt

Im Zeitschriftenarchiv werden die Dokumente über Konnektoren mit den einzelnen Wissensquellen, die als gelbe Kreise mit gezacktem Rand dargestellt werden, zugewiesen, damit sich im Modell folgende baumartige Struktur ergibt:



**Abbildung 41: Strukturiertes Dokumentarchiv mit Wissensquellen**

Das Objekt *Topic* steht für ein Thema. Neben dem Namen ist vor allem das Attribut „Referenzierte Wissensressourcen“ wichtig, welches auf beliebig viele Dokumente verweisen kann. Diese sind im Dialogfenster Notebook des Themenobjekts aufgelistet, wie die nächste Abbildung zeigt:

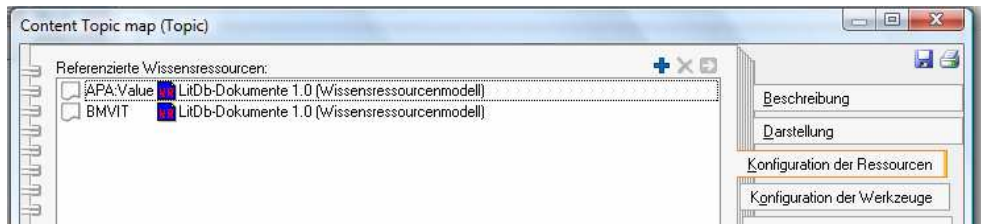


Abbildung 42: Referenzierte Dokumente eines Topics

Umgekehrt betrachtet kann ein Dokument einem oder mehreren Themen (Topics) zugeordnet werden, die es in einer Themen- bzw. Schlagwortsuche auffindbar machen. Hierfür wird eine Themenlandkarte (Topic Map) in einem Wissensstrukturmodell abgebildet, die die einzelnen Themen hierarchisch strukturiert, wie die nächste Grafik exemplarisch zeigt:

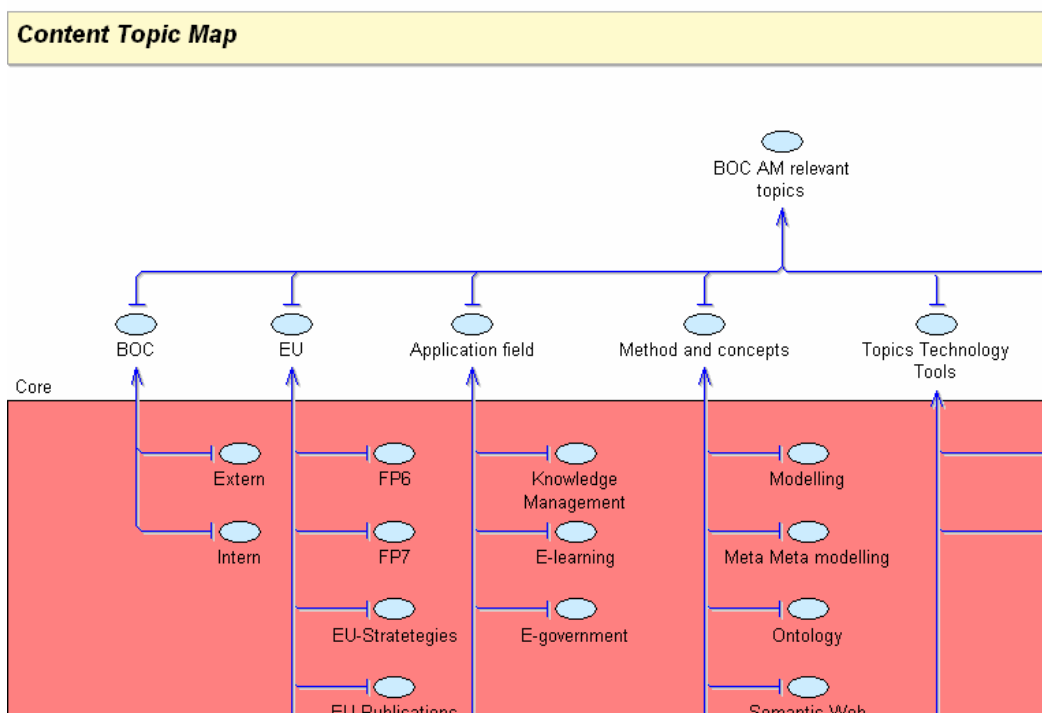


Abbildung 43: Ausschnitt einer Topic Map zur Annotation von Dokumenten

Die Daten und Beziehungen dieser Modelle, wie z. B. der Name eines Objekts oder eine Verbindung zwischen Objekten, werden vom Webframework ausgelesen. Der Modell Editor und die Modell Annotation besitzen wiederum die Möglichkeit den Modellbestand zu verändern. Zur Gewährleistung dieser Funktionalität werden eine Benutzeroberfläche und eine Geschäftslogik implementiert, die ausschließlich auf der Programmiersprache Java basieren. Die Architektur wird in Form von UML Klassendiagrammen im nächsten Abschnitt beschrieben.

## 4.2.3 Klassendiagramme

Dieses Kapitel beschreibt die Java Klassen, die auf den Bibliotheken der Geschäftslogik (AdoWebservice Client, AdoScript Object Library und AdoScript Objects Event Delegate) und der Benutzeroberfläche (RAP Widget Library, RAP Widget Event Delegate, Suchmaschine, Modell Editor und Modell Annotation) entwickelt werden sollen. Im Zuge dessen sollen vier Pakete (packages) realisiert werden:

- Basispaket (com.boc\_eu.adoweb.component.litdb)
- AdoScript Objects (com.boc\_eu.adoweb.component.litdb.aso)
- RAP Benutzeroberfläche (com.boc\_eu.adoweb.component.litdb.ui)
- Helferklassen (com.boc\_eu.adoweb.component.litdb.util)

Das Basispaket, welches für die Initialisierung der Applikation zuständig ist, hat folgendes Aussehen:

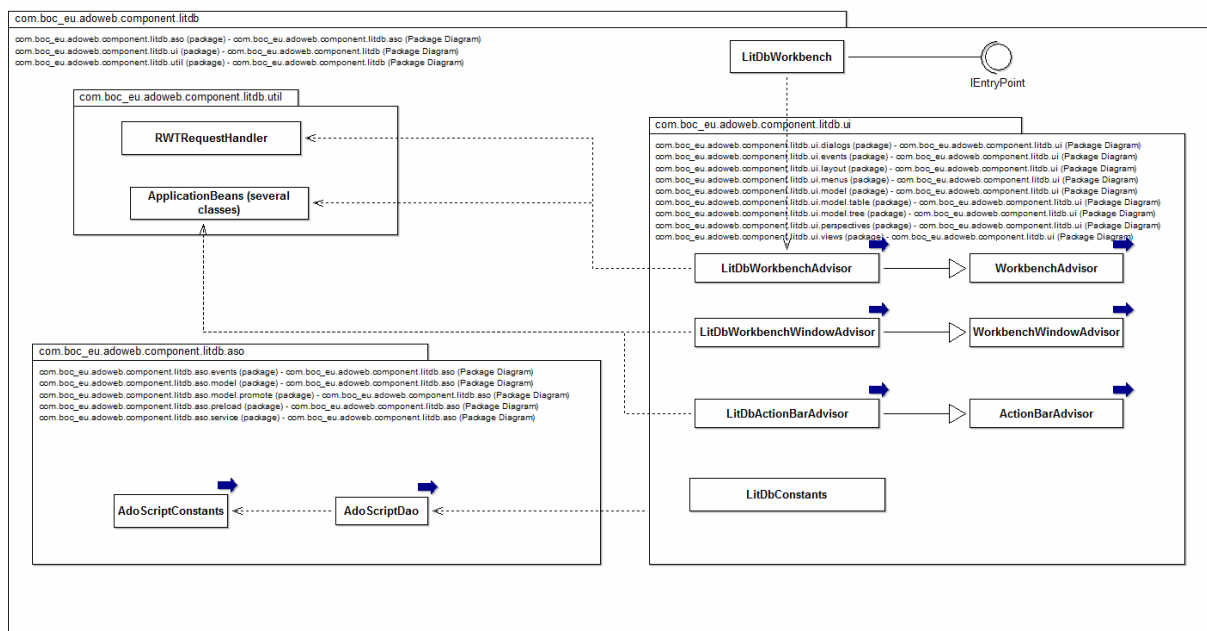
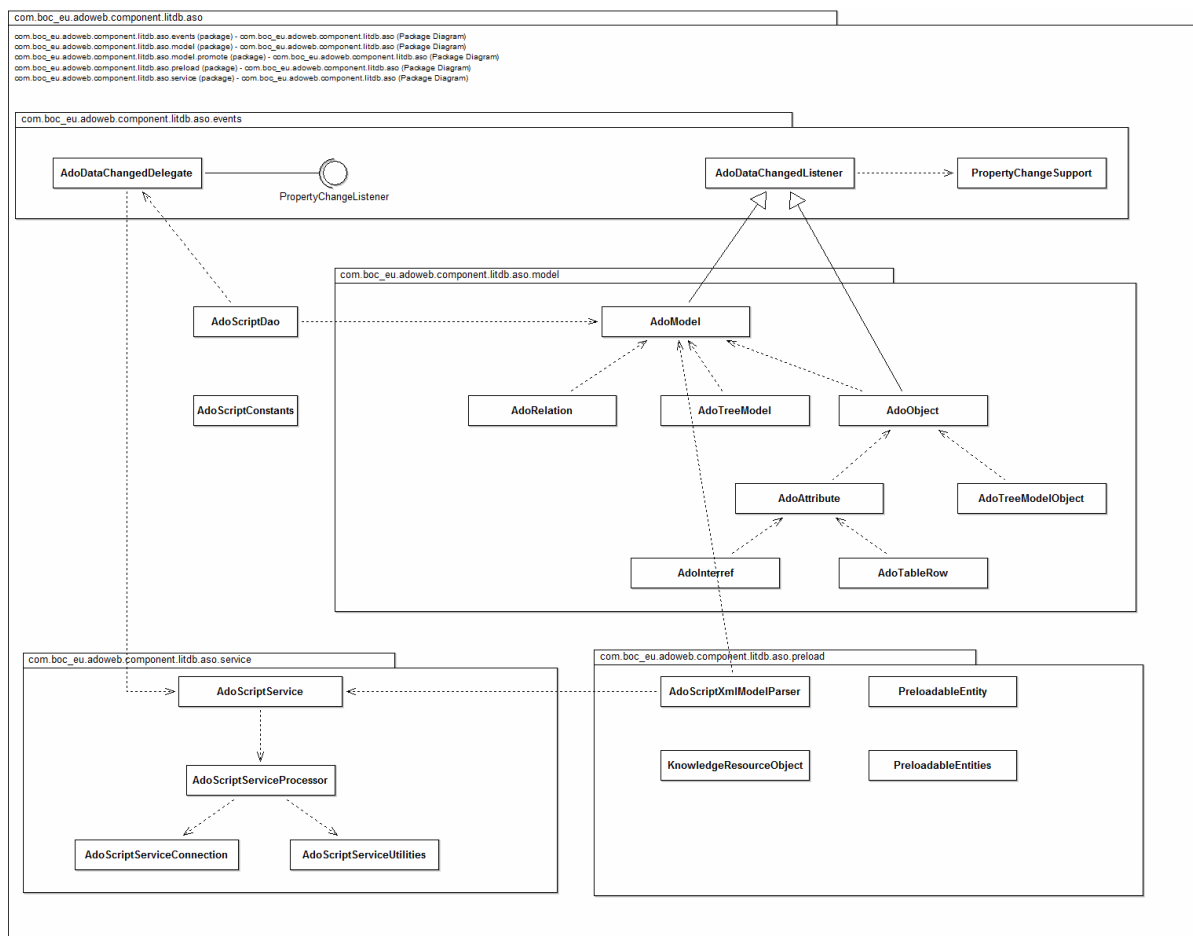


Abbildung 44: Basispaket (com.boc\_eu.adoweb.component.litdb)

Durch Eingabe einer definierten URL (siehe Kapitel 4.3) im Browser wird die Applikation auf Anfrage des Benutzers gestartet. Die HTTP Anfrage wird über das RAP Bridge Servlet direkt an die Java Klasse LitDbWorkbench weitergeleitet, die das Interface IEntryPoint mit der Methode public Display createUI() implementiert. Diese Methode gibt die Zeichenfläche (Display) zurück, auf die verschiedene Perspektiven (Ansichten), Views (Fenster), Widgets (Benutzeroberflächenelemente) und Events (Ereignisse) registriert werden können. Die

Perspektive wird von der Klasse LitDbWorkbenchAdvisor mit der Utility RWTRestHandler, die auf bestimmte URL Parameter reagiert, aufgerufen. Der LitDbWorkbenchWindowAdvisor konfiguriert das Aussehen der Workbench, während der LitDbActionBarAdvisor die Ereignisse auf die Menueinträge setzt. Die GUI Klassen im com.boc\_eu.adoweb.component.litdb.ui Paket benutzen die gemeinsame Schnittstelle AdoScriptDao, die die Daten aus dem AdoScriptService verwaltet, wie die folgende Abbildung der AdoScript Objects com.boc\_eu.adoweb.component.litdb.aso Pakets zeigt:



**Abbildung 45: AdoScript Objects (com.boc\_eu.adoweb.component.litdb.aso)**

Die AdoScriptDao Klasse registriert verschiedene Modelle über die AdoModel Klasse, die entweder ein neues Modell anlegt oder bestehende Modelle über den Namen, Typ und Version abrufen kann. Zur Abfrage der Modellinformationen wird im Konstruktor der AdoModel Klasse der AdoScriptXmlModelParser ausgeführt, welcher über den AdoScriptService eine XML Repräsentation eines ADONIS Modell erhält und anschließend in die grundlegende AdoScript Objects Datenstruktur, d. h. AdoModel, AdoObjects, AdoRelation, AdoInterref, AdoTableRow und AdoAttribute, einliest. Die Klassen AdoTreeModel und AdoTreeModelObject sind spezielle Ausprägungen, die aus AdoObjects



und `AdoRelations` eine hierarchische Struktur abbilden, die für die Darstellung eines Baum Widgets eingesetzt werden kann. Die `AdoScriptObjects` Klassen aus dem Paket `com.boc_eu.adoweb.component.litdb.aso.model` leiten sich von der Klasse `AdoDataChangeListener` ab, das ein Objekt enthält, welches ein Ereignis auslösen kann. Dieses stößt den `AdoDataChangedDelegate` an, der zuvor von der `AdoScriptDao` registriert wurde. Dem Delegat wird das auslösende Objekt übergeben, um den dazugehörigen `AdoScript` Befehl über den Service aufzurufen. Neben der abstrahierenden Logik besitzt dieses Paket die Funktion eines Datenmodells und Cache (Zwischenspeicher).

Das letzte und bei weitem umfangreichste Paket ist die RAP Benutzeroberfläche (`com.boc_eu.adoweb.component.litdb.ui`). Im Basispaket wird nur eine Perspektive aufgerufen, die zum einem in der globalen Konfigurationsdatei `plugin.xml` registriert wird, und zum anderen in einer Java Klasse die Referenzen auf die Views (Fenster) beinhaltet. Im Falle der Literaturdatenbank gibt es zwei Perspektiven:

- Browse Perspective (Modell Editor & Annotation)
- Search Perspective (Suchmaschine)

Zwei Views (`LitDbMagazineTreeViewPart` und `LitDbBlogTableViewPart`) besitzen explizite Kontextmenüs, auf die verschiedene Ereignisse des `com.boc_eu.adoweb.component.litdb.ui.events` Pakets registriert sind. Diese lösen bei Bedarf wiederum Fensterdialoge, wie z. B. `NewObjectDialog`, aus, die den Benutzern kurze Rückmeldungen bzw. `Minimaldialoge` anzeigen. Die Views bestehen aus Widgets, die mit Werten aus dem `com.boc_eu.adoweb.component.litdb.ui.model` Paket befüllt werden können. Die `JFace ContentProvider` bereiten die Daten für die Darstellung in Tabellen und Bäumen auf. Diese beziehen sie aus der Schnittstelle `AdoScriptDao`, deren Funktionalität in der vorherigen Abbildung erläutert wurde. Abschließend sei noch ein selbst definiertes Layout für den `LitDbEditorViewPart`, welches die Breite der Widgets, in diesem Fall Labels und Textfelder, automatisch an die Breite der View anpasst und sie zeilenweise anordnet. Im folgenden Kapitel wird die Benutzersicht der gesamten Applikation, die die Softwarebibliotheken, das Modell Repository und die Klassendigramme umfasst, erläutert, während die nächste Abbildung die komplexe Struktur des Benutzeroberflächenpakets veranschaulicht.



### 4.3 Die Workbench

Der Betrieb des Prototyps benötigt einen Client und einen Server, die über ein gemeinsames Netzwerk miteinander verbunden sind. Die ausgelieferte Webapplikation wird in einer Server Umgebung installiert, wodurch sie für den Client über eine eindeutige URL aufrufbar wird. Des Weiteren ergeben sich auf der Serverseite folgende Voraussetzungen:

- Windows 2000/XP/Vista
- MSDE SQL Server 2000
- MSDE Datenbank
- ADONIS 3.9 Deutsch
- PROMOTE Anwendungsbibliothek 1.4 beta 070727 (ABL)
- Initiale PROMOTE Modelle (ADL)
- In ADONIS inkludierte AdoScript InitXML und DocuXML Dateien (ASC)
- Java Runtime Environment 1.5 oder höher
- Ein Servlet Container, empfohlen Apache Tomcat 5.5.20
- Die Ajax Literaturdatenbank als Webapplikation (WAR)

Außerdem muss im ADONIS Administrationstoolkit ein Benutzer angelegt werden. Dann wird ADONIS als Webservice Instanz auf einem Port gestartet, womit der AdoScriptService über HTTP ansprechbar wird. Das Webframework besteht aus der Java Runtime, dem Servlet Container Tomcat und der Webapplikation, die ihrerseits installiert werden müssen. Die einzige Konfiguration besteht im Definieren des Ports zu ADONIS, die in der Property „loadbalancer.endpoints“ der Datei „context.xml“ des litdb.war Archivs zu erfolgen hat. Das Webframework ist auf der beiliegenden CD enthalten. Die Zugabe der anderen Bestandteile ist aus Lizenzgründen nicht möglich.

Die Anforderungen der Clientseite sind stark von den eingesetzten Ajax Frameworks RAP 1.0 und Qooxdoo 0.7 geprägt. Neben einer gewöhnlichen DSL bzw. Breitbandverbindung hat die Anwendung folgende Abhängigkeiten:

- Windows, Linux oder Mac OS X Betriebssystem
- Ein Webbrowser:
  - Internet Explorer 5.x oder höher

- Mozilla Firefox 1.0 oder höher
- Aktiviertes JavaScript
- Erlaubte Cookies

Bei korrekter Installation kann die Suchmaschine unter <http://localhost:8080/litdb/workbenchh?perspective=search> und der Modell Editor und Annotation Service unter <http://localhost:8080/litdb/workbench?perspective=annotator> aufgerufen werden.

Als Nächstes soll ein kurzer Einblick in die Hintergrundprozesse des RAP Frameworks gewährt werden. Danach erfolgt die Vorstellung der zwei Services Modell Editor und Annotation und Suchmaschine, die als Perspektiven der Workbench realisiert wurden.

### 4.3.1 Technologischer Hintergrund

Die Literaturdatenbank (litdb.war) hat über zwei URLs, die die zwei Perspektiven, nämlich Modell Editor & Annotation und Suchmaschine, zeichnen, aufgerufen werden. Das BridgeServlet in der Bibliothek „servletbrige.jar“ im Verzeichnis „WEB-INF/lib“ leitet den Aufruf an den Equinox Server weiter, welcher ein Webframework für Eclipse Plugins ist. Das Equinox Framework liest aus der Datei „config.ini“ im Verzeichnis „WEB-INF/eclipse/configuration“ die zu ladenden Plugins, die sich im Verzeichnis „WEB-INF/eclipse/plugins“ befinden. Die Plugins können eine Konfigurationsdatei namens „plugin.xml“ enthalten, wie in diesem Fall das Applikationsplugin „LitDb v1.2\_1.0.0.20071010-2241.jar“. Dort werden der Name und die Parameter der Servlet Delegaten, die registrierten Perspektiven, Views, Ereignisse und Themes bzw. Skins definiert.

Die Literaturdatenbank verwendet zur Darstellung und zum Ereignismanagement der Benutzeroberfläche die RAP Pakete RWT (org.eclipse.rap.rwt), UI (org.eclipse.rap.ui) und JFace (org.eclipse.rap.jface), welche von den SWT (org.eclipse.swt) und JFace (org.eclipse.jface) Eclipse Bibliotheken abgeleitet sind. Die implementierten Widgets einer Applikation, wie z. B. Textfelder, Tabellen oder Menüs, werden zur Laufzeit in äquivalente Qooxdoo JavaScript Objekte übersetzt. Ein Beispiel hierfür ist ein Button, der in Java RWT mit dem Objekt „org.eclipse.swt.widgets.Button“ programmiert und im Browser mit dem

Objekt „qx.ui.form.Button“ dargestellt wird. Ereignismeldungen, wie z. B. das Drücken eines Buttons, werden über statische JavaScript Methoden realisiert, die Nachrichten an das Serverobjekt schicken. Objektbeziehungen zwischen RWT (Java) und Qooxdoo (JavaScript) werden über die RWT Lebenszyklusphasen realisiert:<sup>142</sup>

- *PrepareUIRoot*: Erzeugen des JavaScript Widgets
- *ReadData*: Server liest Werte, die vom Client gesendet werden
- *ProcessAction*: Das Widget wird über ein Ereignis benachrichtigt
- *Render*: Die clientseitige JavaScript Ausgabe wird aktualisiert, wie z. B. das Auffächern eines Baums

Überzeugend ist auf jeden Fall das Aussehen der Ajax Workbench, die der Eclipse Workbench sehr ähnlich sieht. Sie besteht aus einem frei verschiebbaren Fenster im Browser und weist u. a. folgende Elemente:

- Menuleiste mit beliebig verschachtelbaren Menüs
- Perspektivenfensterleisten, zum Wechseln zwischen Perspektiven
- Toolbar
- Perspektiven
- Views
- Dialogboxen

Die nächsten beiden Abschnitte stellen die beiden Perspektiven Modell Editor und Annotation und Suchmaschine vor, die die drei Services der Literaturdatenbank bereitstellen.

### **4.3.2 Modell Editor und Annotation**

Die Services Modell Editor und Modell Annotation werden aufgrund von Usability Gründen zu einer Perspektive zusammengefasst, die folgendes Aussehen hat:

---

<sup>142</sup> Appel, Frank u. Krause, Jochen (2007): „Developing with the Rich Ajax Platform“, S. 15

Menüleiste

Perspektiven Wechsler

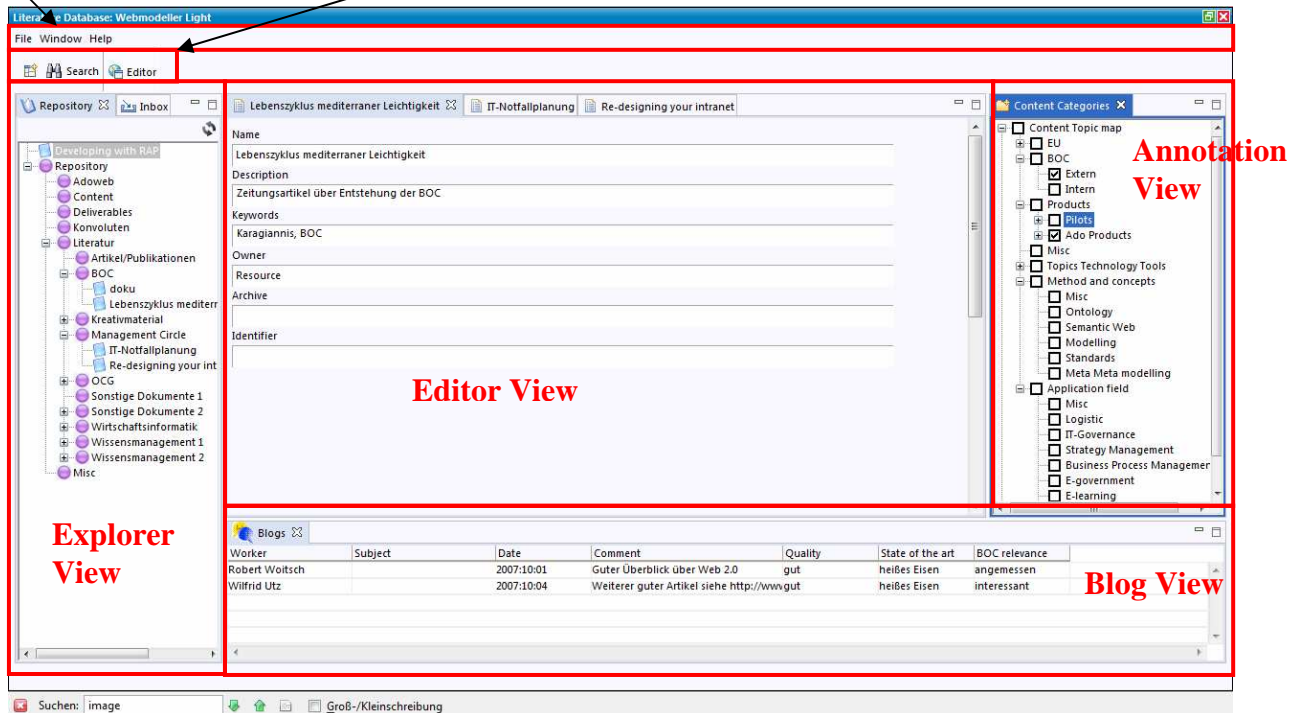


Abbildung 47: Modell Editor & Annotation Perspektive

Die Explorer View (Explorer Fenster) auf der linken Seite stellt das Literatur Archiv, welches die Wissensquellen als Elternordner bzw. die Dokumente als dessen Kinder in einer baumartigen Struktur abbildet. Die Struktur wird aus dem Dokumentenarchiv des Modell Repository (siehe Abbildung 41) übernommen. Die Transformation in diese hierarchische Darstellung ist recht kompliziert. Das Modell wird zuerst in einer XML-Repräsentation ausgelesen, dann werden das Modell und dessen Objekte in ein AdoModel Objekt bzw. in AdoObject Objekte verpackt. Danach erfolgt eine weitere Transformation zu einem AdoTreeModel Objekt bzw. zu AdoTreeModelObject Objekten, die auf Basis der Konnektoren (gerichtete Pfeile) des Modells eine baumartige Struktur rekursiv aufbauen. Zum Schluss erfolgt die eigentliche Darstellung als RWT Baum mit Knoten, die wiederum eine Rekursion benötigen. Die Lösung ist aufgrund der doppelten Rekursion nicht performant. Dennoch ist die Darstellung des Baums und das damit verbundene Ereignis Handling gelungen, welches mit Hilfe eines Kontextmenüs Aktionen für einzelne Knoten auslösen kann:

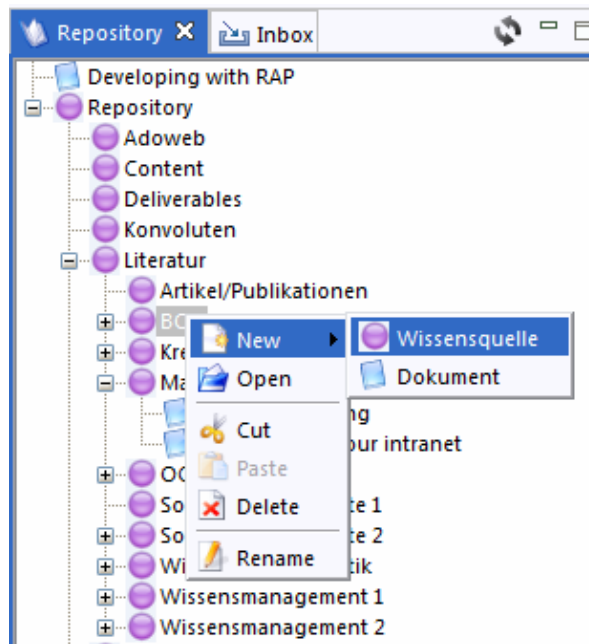


Abbildung 48: Explorer View des Literatur Repository

Die Wissensquellen werden in der Abbildung als violette Kreise und die Dokumente als Bücher dargestellt. Das Kontextmenü, welches mit einem rechten Mausklick aktiviert wird, beinhaltet insgesamt sechs Aktionen, die abhängig von der Markierung im Baum sind:

- *New*: Anlegen einer Wissensquelle oder eines Dokuments
- *Open*: Öffnen der Detailansicht einer Wissensquelle
- *Cut*: Ausschneiden eines Knotens
- *Paste*: Einfügen eines zuvor ausgeschnittenen Knotens
- *Delete*: Löschen eines Knotens
- *Rename*: Umbenennen eines Knotens

Wenn nun ein neues Dokument angelegt werden soll, öffnet die Markierung des Menüeintrags „New“ ein Untermenü mit den Einträgen „Wissensquelle“ bzw. „Dokument“. Das Klicken auf „Dokument“ öffnet folgenden Dialog:

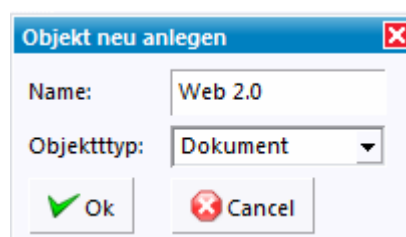
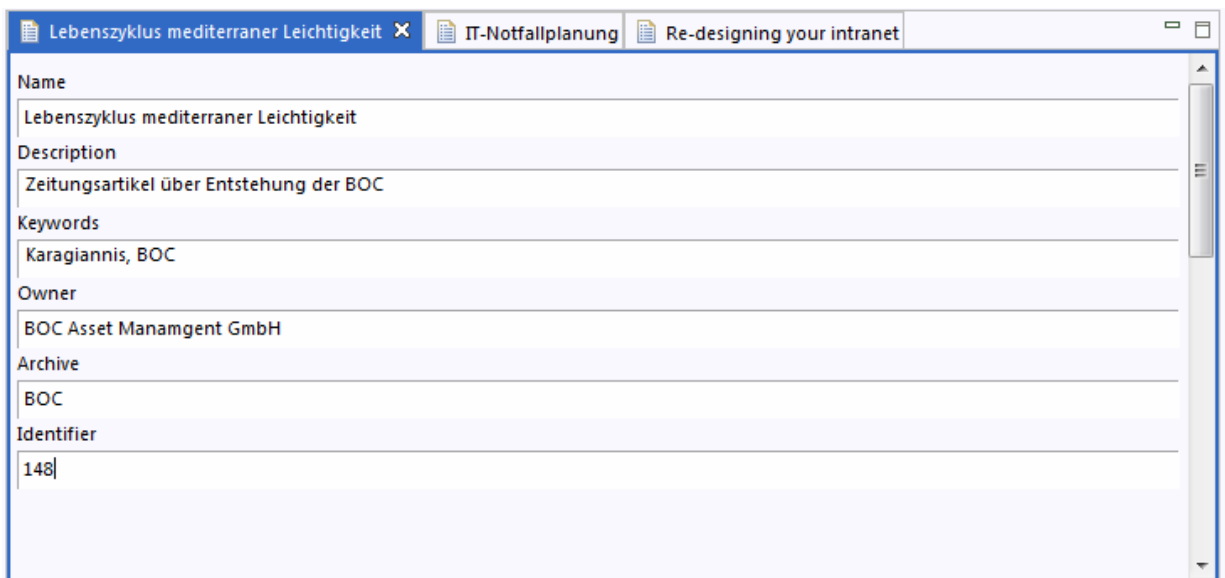


Abbildung 49: Neues Dokument anlegen

Der Dialog erfordert das Eingeben eines Namens. Danach wird der zuvor ausgegraute Button „Ok“ aktiviert. Im Falle einer Bestätigung wird ein Kindknoten des markierten Elternknotens im Baum eingefügt. Die Prozedur für das Anlegen einer Wissensquelle verhält sich äquivalent.

Das Öffnen einer Ressource zur Detailansicht erfolgt entweder über einen Doppelklick auf den gewünschten Baumknoten bzw. über das Auswählen der „Open“ Aktion im Kontextmenü. Daraufhin wird in der Editor View der Perspektive ein neuer Reiter geöffnet, der die Attribute eines Objekts anzeigt, wie die nächste Abbildung darstellt:



**Abbildung 50: Detailansicht eines Dokuments**

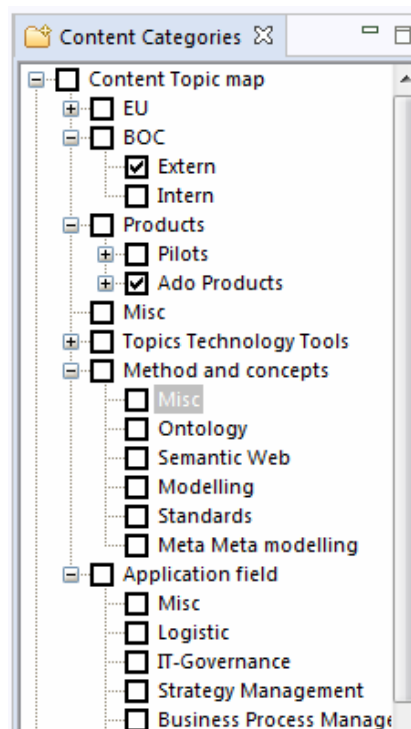
Das Ändern eines Wertes in einem Textfeld wird sofort in den Modellbestand übernommen, Dazu wird jedem Textfeld ein ModifyListener zugewiesen, der jegliche Änderung automatisch an den AdoScript Objects Delegate weitergibt und für die Synchronisierung mit dem Modellbestand verantwortlich ist. Es können auch mehrere Dokumente bzw. Wissensquellen parallel editiert werden, da das Tabbing eine multiple Ansicht ermöglicht.

Das Ausschneiden bzw. Einfügen eines Knotens entspricht im Modellbestand dem Löschen eines alten und dem Anlegen eines neuen Konnektors. Dazu wird der gewünschte Knoten markiert und mit der „Cut“ Aktion des Kontextmenüs ausgeschnitten. Die Aktion „Paste“ fügt den Knoten als Kind des neu markierten Elternknotens wieder ein. Als Hilfestellung werden die Einträge „Cut“ und „Paste“ nach Ausführen einer Aktion ausgegraut.



Das Umbenennen eines Knotens öffnet einen ähnlichen Dialog wie das Erstellen eines Knotens. Die Eingabe des neuen Namens ist erforderlich, was durch Drücken des „Ok“ Buttons bestätigt wird.

Die Annotation eines Dokuments erfolgt durch die äußerst rechte Annotation View der Abbildung 47, welche abhängig vom geöffneten Dokument in der Editor View ist. Die Annotation View hält eine Themenlandkarte, dessen Transformation aus dem Modellbestand sich analog zum Literaturarchiv verhält, wie die Grafik zeigt:



**Abbildung 51: Themenlandkarte in der Annotation View**

Zusätzlich besitzt diese Baumansicht Checkboxen, die anzeigen, welche Topics auf dieses Dokument referenzieren. Durch An- bzw. Abwählen werden diese Verweise angelegt bzw. gelöscht (siehe Abbildung 42).

Die Blog View ist nur ein Showcase, da sie nicht mit dem Modell Repository gekoppelt ist. Sie hat eine Tabelle, die eine Liste von Benutzereinträgen enthält. Ein Eintrag enthält den Namen, Titel, das Datum und ein Kommentar. Des Weiteren kann noch eine Bewertung abgegeben werden, die sich auf das geöffnete Dokument bzw. Wissensquelle im Editor View bezieht. Die Blog View mit der Tabelle beschreibt der folgende Screenshot:

Worker	Subject	Date	Comment	Quality	State of the art	BOC relevance
Robert Woitsch		2007:10:01	Guter Überblick über Web 2.0	gut	heißes Eisen	angemessen
Wilfrid Utz		2007:10:04	Weiterer guter Artikel siehe <a href="http://www.wvlgut">http://www.wvlgut</a>	gut	heißes Eisen	interessant

Abbildung 52: Blog View

Die Tabelle besitzt ein Kontextmenü, mit dem ein neuer Blogbeitrag gepostet werden kann. Bei Auswahl „New entry“ öffnet sich folgendes Dialogfenster:

Abbildung 53: Neuer Blog Eintrag

Der Explorer View enthält noch einen weiteren Reiter, nämlich die Inbox. Diese ist dazu gedacht, neu eingestellte Dokumente anzuzeigen, die alle Wissensressourcenmodelle enthält, die in ADONIS in der Modellgruppe „Acquisition“ angelegt werden. Die nächste Abbildung zeigt dies beispielhaft anhand von drei Dokumenten:

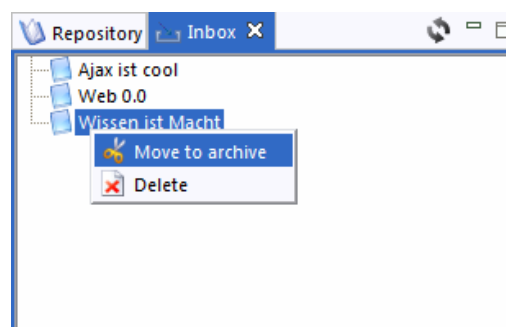


Abbildung 54: Literatur Inbox View

Die Einstellung erfolgt über ein drittes Service, nämlich der Excel Akquisition, die bereits Bestandteil des ADOweb Service Frameworks ist und in dieser Arbeit nicht weiter ausgeführt wird. Das Kontextmenü bietet hier die Funktion „Move to archive“, mit der das Dokumentobjekt in das dauerhafte Repository verschoben werden kann. Darüber hinaus kann das Dokument auch mit „Delete“ gelöscht werden. Anzumerken ist der „Refresh“ Button in der rechten oberen Ecke des Views, welches Daten aus dem Modellbestand aktualisiert. Damit können Änderungen am Modellbestand, die nicht über die Ajax Literaturdatenbank, wie z. B. das Editieren im Webmodeller, erfolgt sind, übernommen werden.

Der Modell Editor und Annotation haben gezeigt, wie Dokumente eingestellt, verändert, gelöscht und annotiert werden können. Des Weiteren kann der Modellbestand durchgeblättert und Detailinformation von Objekten angezeigt werden. Das gezielte Auffinden von Dokumenten wird durch den Wissensnutzungsprozess der Suchmaschine ermöglicht, der im nächsten Abschnitt behandelt wird.

### 4.3.3 Suchmaschine

Der Service der Suchmaschine dient dazu Dokumente, die sich in der Literaturdatenbank befinden, aufzufinden. Sie ist um einiges einfacher gehalten als der Modell Editor und die Annotation, da sie auch von Benutzern verwendet werden soll, die kein Wissen über Modellierung besitzen. Die Perspektive der Suchmaschine hat folgendes Aussehen:

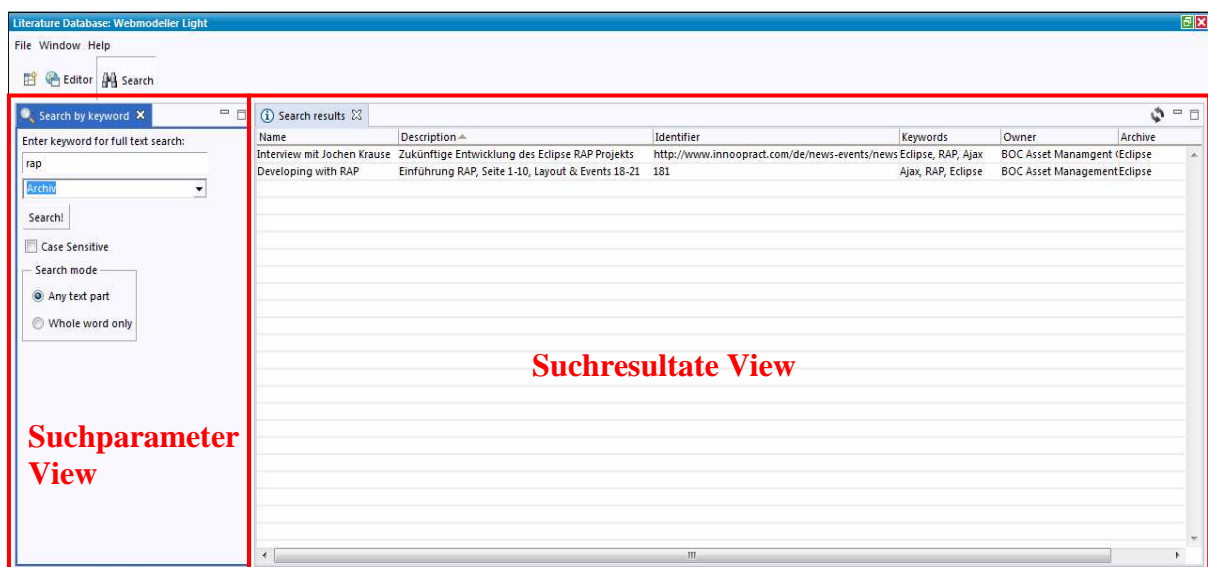


Abbildung 55: Perspektive der Suchmaschine

Die Suchmaschine besitzt zwei Views, nämlich das Suchparameter View und das Suchresultate View, wovon die erstere zunächst erläutert wird:

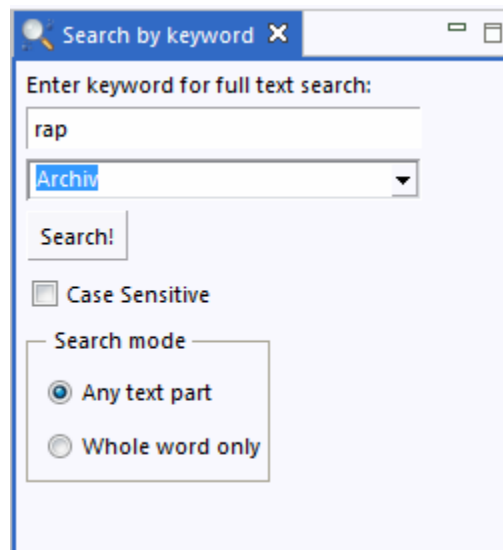
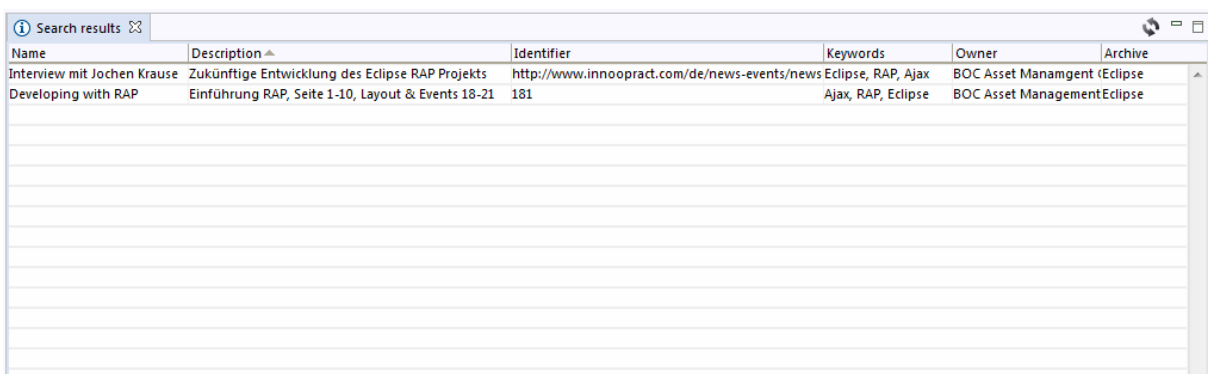


Abbildung 56: Suchparameter View

Im Suchparameter View wird das Schlagwort spezifiziert, das in den Attributen der Dokumente gesucht werden soll. Unter dem Textfeld befindet sich der Suchort. Zurzeit gibt es zwei Orte, nämlich das Archiv oder den Eingang. Diese Unterteilung hat den Zweck kürzlich eingestellte Dokumente sofort zu sehen, damit sie nicht im Archiv untergehen. Unter dem „Search“ Button kann noch die Suchart des Schlagworts definiert werden. Ein Anklicken der „Case Sensitive“ Checkbox bewirkt eine Berücksichtigung der Groß- bzw. Kleinschreibung. Der Suchmodus „Any text part“ sucht nach allen möglichen Text- bzw. Wörtern und „Whole word only“ sucht nur nach ganzen Wörtern. Wenn der „Search“ Button gedrückt wird, gibt das Suchresultate View die gewünschten Suchresultate preis:



Name	Description	Identifier	Keywords	Owner	Archive
Interview mit Jochen Krause	Zukünftige Entwicklung des Eclipse RAP Projekts	<a href="http://www.innoo pract.com/de/news-events/news">http://www.innoo pract.com/de/news-events/news</a>	Eclipse, RAP, Ajax	BOC Asset Managment (Eclipse	
Developing with RAP	Einführung RAP, Seite 1-10, Layout & Events 18-21	181	Ajax, RAP, Eclipse	BOC Asset ManagementEclipse	

Abbildung 57: Suchresultate View

Wenn das Suchwortfeld leer gelassen wird, werden alle Dokumente ausgegeben. In diesem Beispiel wurde das Suchwort „rap“ eingegeben, dass zwei Artikel zurückliefert, nämlich ein Interview mit Jochen Krause und ein Entwicklungsdokument über RAP. Es handelt sich um eine Volltextsuche, d. h. alle Attribute eines Objekts werden nach Übereinstimmungen durchsucht. In der rechten, oberen Ecke befindet sich wieder ein „Refresh“ Button, so dass auf Änderungen im Modellbestand reagiert werden kann.

Die letzten beiden Abschnitte haben die Funktionalitäten erläutert, die dem Benutzer zur Verfügung stehen. Die Implementierung dieser war jedoch nicht ganz unproblematisch, wovon im nächsten Abschnitt die Rede ist.

#### **4.4 Einschränkungen und Probleme**

Der Implementierungsprozess ähnelte eher einem „Trial und Error“ als einer methodischen Vorgehensweise. Der Grund lag einerseits in der fast nicht vorhandenen Dokumentation, andererseits in den unvollkommenen Beta Versionen von RAP, die nicht alle Funktionen richtig unterstützten. Die Entwicklung begann beim Milestone 3 (M3) Release von insgesamt sechs Milestones. Die Portierung zwischen den Milestone Versionen lief jedoch ohne Komplikationen, was einem guten, stabilen Grundkonzept zu verdanken ist. RAP machte von M3 bis zur Version 1.0 merkliche Fortschritte und jede Version war einer Aktualisierung würdig. In der Version 1.0 wurden viele Performanz- und die Stabilitätsprobleme zu einem großen Teil behoben, was für einen professionellen, kommerziellen Einsatz spricht.

Eine RAP Applikation startet erst auf Anfrage für jeden Benutzer separat. Sie wird zwar vom unterliegenden Servlet Container initialisiert, jedoch können keine Daten vorgeladen werden. Dies erscheint vor allem dann als Nachteil, wenn es Daten geben soll, die für alle Benutzer gleich sind, wie z. B. Dokumente eines Zeitschriftenarchivs. Das Fehlen jeglicher Persistenz schlägt sich bei vielen konkurrenten Zugriffen einerseits schlecht auf die Performanz nieder, andererseits kann dies zu erheblichen Speicherproblemen führen, wenn der Cache entsprechend groß ist. Abgesehen davon muss man sich bei RAP auf den generierten Code verlassen können. Da Qooxdoo auch noch keine Version 1.0 hat, lassen sich nicht alle Fehler dem RAP Framework zuschreiben. Die Klassenvererbung und Objektorientierung von Qooxdoo weist ähnliche Züge wie das Eclipse SWT aus. Jedoch handelt es sich immer noch

um unterschiedliche APIs, die aufeinander abgestimmt werden müssen. Nichtsdestotrotz scheinen die Entwickler von RAP und Qooxdoo, die übrigens beide Unternehmen aus Deutschland sind, sehr gut zusammen zu arbeiten. In Zukunft wird man voraussichtlich noch sehr viel Positives über RAP hören können.

Schätzungsweise 80 % der Fehler, die in der Version 1.2 der Literaturdatenbank auftreten, finden ihren Ursprung in der Implementierung. Teilweise werden auf der Serverseite Java vor allem NullPointerExceptions (Ausnahmen durch Zuweisung einer Variable mit dem „null“ Wert) nicht abgefangen, mit denen JavaScript nicht umgehen kann. Darüber hinaus werden auch nicht alle GUI Funktionalitäten wie gewünscht unterstützt, wie z. B. das Anlegen von Annotationen in der Annotation View. Der Grund ist ein Zeitmangel. Auch andere Widgets machen nicht immer das, was man ihnen in Java zu sagen versucht, jedoch ist die Qualität des Codes dennoch zufrieden stellend.

Das Projekt umfasst 23 Packages, 77 Klassen und 8507 Zeilen Code. Im eigentlichen Sinne sind es ja zwei Projekte, nämlich das RAP GUI und das Datenmodell der AdoScriptObjects Bibliothek, darin enthalten. Eine Trennung der Projekte bzw. eine saubere Trennung von Darstellung und Geschäftslogik ist in den weiteren Versionen unbedingt erforderlich.

Es wurde übermäßig viel Zeit für die Entwicklung des Datenmodells (AdoScriptObjects Bibliothek) aufgewendet. Dies wurde deswegen investiert, da es sich um eine Bibliothek handelt, die die Entwicklung um ein Vielfaches beschleunigt. Sie wurde mit einigen Adaptionen in anderen Projekten, wie z. B. dem TopicMapFileSystemExport, die eine Themenlandkarte zu einem Dateisystem exportiert, oder dem FileSystem Observer, der Dateien in Form von Dokumenten in den Modellbestand importiert, erfolgreich eingesetzt. Diese Programme bzw. Services konnten innerhalb weniger Tage entwickelt werden, da man das Datenmodell wieder verwenden und sich auf die Programmierung der eigentlichen Applikationsfunktionalität konzentrieren konnte. Eine Erweiterung auf andere Datenquellen, wie z. B. einer Datenbank, Dateien oder anderen Scriptsprachen, wie z. B. JavaScript, ist nicht nur für die Plattform ADOxx interessant. Die AdoScriptObjects Bibliothek ist als Client realisiert, womit sie Cache und Zugriffslogik zugleich ist. Dies steht jedoch etwas im Gegensatz zu ADOweb, welches diese Logik in den Services ReadData und WriteData realisiert. Eine Vereinigung beider Ansätze ist vorzuschlagen. Ein Vorschlag wäre, die Objekte der AdoScript Objects zu serialisieren, um sie als Rückgabewert eines Webservice zu

übergeben. Der Client benötigt dann nur mehr einen „Stub“, welches die Zugriffslogik auf die Schnittstelle repräsentiert, und muss nicht mehr die gesamte Bibliothek halten. Diese sollte im ADOweb Service Framework realisiert werden.

Die Abspeicherung im Modellbestand läuft zwar reibungslos, jedoch werden die neuen Objekte grafisch an den Koordinaten (0,0) abgelegt, was bedeutet, dass alle Objekte übereinander liegen. Dies ist eine Tatsache und kein Fehler. Zur Behebung dieser Unzulänglichkeit wird ein grafischer Positionierungsalgorithmus benötigt, der freie Plätze auf der Zeichenfläche sucht und die Objekte systematisch abspeichern kann.

Das Framework RAP unterstützt zwar viele Funktionen einer Rich Internet Applikation, jedoch fehlt z. B. eine Möglichkeit zum File Upload. Eine weitere Funktion, die von RAP nicht bereit gestellt wird, ist ein Mausereignismelder, der die Koordinaten eines Mausklicks liefert. Dazu musste im Zuge des Projekts die Plattform erweitert werden. Der nächste Abschnitt erklärt diesen Eingriff und gewährt gleichzeitig interessante Einblicke in das Wesen der RAP Plattform.

#### **4.5 Exkurs: Mouse Listener**

Im Zuge der Entwicklung auf dem Milestone 3 Release stellte sich aufgrund von fehlender Unterstützung des RAP Frameworks folgende Aufgabe. Die Darstellung von großen Grafiken wurde mit einem Label Objekt realisiert. Zum Abfangen von Ereignissen war es interessant, an welcher Position in der Grafik ein Mausereignis ausgelöst wurde, um ein Objekt innerhalb des Bildes selektieren zu können. RAP ist zwar in der Lage, Selektionsereignisse mit der linken Maustaste bzw. Kontextmenüereignisse mit der rechten Maustaste abzufangen, jedoch nicht deren X,Y Koordination zu bestimmen. Dazu bedurfte es Anpassungen in der Plattform, die zwar mittlerweile gut verwendet werden konnte, jedoch über dessen Aufbau kein Verständnis vorhanden war. Deshalb wurde an die Newsgroup von RAP eine Support Anfrage gesendet. Innerhalb weniger Tage meldete sich Ralf Sternberg vom RAP Entwicklungsteam, der sehr hilfsbereit durch diesen Prozess führte. Er übermittelte seinen Code über die Erweiterung eines Label Widgets mit der Bezeichnung „ClickLabel“, welches nichts anderes als ein Label ist, welches auf Mausereignisse der linken Maustaste hört. Der Quellcode des ClickLabels war zwar gut strukturiert, jedoch konnte immer noch nicht der

Zweck bzw. das Zusammenspiel der Einzelteile eruiert werden. Deshalb wurde Mitte Juni 2007 eine weitere informelle Anfrage an die RAP Newsgroup gesendet<sup>143</sup>:

Gottfried Nindl schrieb:

Unfortunately I do not know how the the whole thing works, so I can't extend it...

I know the ClicklabelLCA does the rendering of the JavaScript with the JSWriter...

The JSWriter calls somehow the LabelUtil.js, where I define my Qx MouseEvent???

I would appreciate just some tips!

Thanks a lot!

Er antwortete am Tag darauf:<sup>144</sup>

Gottfried,

just some hints:

\* write a JavaScript util class that contains a click listener, like this:

```
my.LabelUtil.clicked : function( evt ) {
  if( !org_eclipse_rap_rwt_EventUtil_suspend ) {
    var wm = org.eclipse.swt.WidgetManager.getInstance();
    var id = wm.findIdByWidget(this);
    var req = org.eclipse.swt.Request.getInstance();
    req.addEvent( "my.clickEvent", id );
    req.addParameter( "my.click.x", evt.getClientX() );
    req.addParameter( "my.click.y", evt.getClientY() );
    req.send();
  }
}
```

\* register the JavaScript as explained in the thread before this one

\* In the LCA, add this listener to the "click" event (instead of JSConst.JS\_WIDGET\_SELECTED)

\* In the readData method of the LCA, handle the event on your own instead delegating to ControlLCAUtil.processSelection().

I'm sorry, but if you want to extend RAP, you have to figure out by yourself how all the parts play together. You might want to have a look at the RAP intro [1] and the qooxdoo API docs [2], e.g. for the qooxdoo MouseEvent.

---

<sup>143</sup> <http://dev.eclipse.org/newlists/news.eclipse.technology.rap/msg00716.html>

<sup>144</sup> <http://dev.eclipse.org/newlists/news.eclipse.technology.rap/msg00717.html>



Hope this helps,  
Ralf

- [1] <http://wiki.eclipse.org/index.php/WidgetToolkit>
- [2] <http://demo.qooxdoo.org/0.7/apiviewer/#qx.event.type.MouseEvent>

Das war genügend Information, mit dem die Erweiterung implementiert werden konnte. Eine Woche intensives Arbeiten reichten für die Umsetzung des Mouse Listeners. Am Anfang stand die JavaScript Klasse, die das Ereignis des Benutzers im Browser abfangen sollte. Das bedeutet, dass jedes Mal, wenn der Benutzer auf das Label klickt, soll das System von der relativen Position des Mausklicks in Kenntnis gesetzt werden. Dazu wurde eine eigene JavaScript Klasse namens „org.eclipse.rap.qx.MouseEventUtil“ geschrieben, die nichts anderes tut, als das Mausereignis abzufangen, und die gewünschten Informationen an den Server zu senden. Die nächsten Codeausschnitte zeigen dies exemplarisch anhand eines Doppelklicks:

```
qx.Class.define("org.eclipse.rap.qx.MouseEventUtil", {  
    statics : {  
        mouseDoubleClick : function( evt ) {  
            var widgetManager =  
                org.eclipse.swt.WidgetManager.getInstance();  
            var id = widgetManager.findIdByWidget( evt.getTarget() );  
            var event = "org.eclipse.rap.events.mouseDoubleClick";  
            var pageX = evt.getPageX();  
            var pageY = evt.getPageY();  
            var screenX = evt.getScreenX();  
            var screenY = evt.getScreenY();  
            var button = evt.getButton();  
            org.eclipse.rap.qx.MouseEventUtil.sendMouseData ( id, event,  
                pageX, pageY, screenX, screenY, button );  
        },  
  
        sendMouseData : function( id, event, pageX, pageY, screenX, screenY,  
            button ) {  
            if( !org.eclipse.rap.rwt.EventUtil.suspend ) {  
                var req = org.eclipse.swt.Request.getInstance();  
                req.addEvent( event, id );  
                req.addParameter( id + ".page.x", pageX);  
                req.addParameter( id + ".page.y", pageY);  
                req.addParameter( id + ".screen.x", screenX );  
                req.addParameter( id + ".screen.y", screenY );  
                req.addParameter( id + ".button", button);  
                req.send();  
            }  
        }  
    }  
}
```

Der Ausschnitt der JavaScript Klasse „org.eclipse.rap.qx.MouseEventUtil“ hat zwei statische Methoden. Die ersten zwei Zeilen der Methode „mouseDoubleClick“ enthalten viel von der RAP-Zauberei:

```
var widgetManager =
    org.eclipse.swt.WidgetManager.getInstance();
var id = widgetManager.findByIdByWidget( evt.getTarget() );
```

Der WidgetManager, welcher alle erstellten JavaScript Widgets enthält, wird mit der Methode getInstance() geholt. Dann folgt in der zweiten Zeile die Auffindung des eigentlichen Widget. Die ID wird vom ausgelösten Ereignis mit der Methode „evt.getTarget()“ erfragt.

Die dritte Zeile weist dem Ereignis einen Namen zu, unter dem es am Server identifiziert und weiter verarbeitet werden kann. Die folgenden Variablen „pageX“, „pageY“, „screenX“ und „screenY“ speichern die absoluten Bildschirmmauskordinaten. Die Variable „button“ hält fest, welche Taste gedrückt wurde, wie links, Mitte oder rechts. Zum Schluss der Methode „mouseDoubleClick“ wird die Methode „sendMouseData“ aufgerufen, die die gesetzten Variablen mit „req.addParameter(...)“ in eine Anfrage inkludiert und mit „req.send()“ an den Server verschickt.

Diese JavaScript Datei wird beim ersten Aufruf des Client an den Server übermittelt und wird nie verändert. Dazu muss sie als Ressource mit einer eindeutig auffindbaren URL registriert werden. Die vereinfachte Klasse leitet sich von der Überklasse „Resource“ bzw. „QooxdooResource“ ab.

```
public class MouseEventResource extends QooxdooResource {
    public String getLocation() {
        return "org/eclipse/rap/qx/MouseEventUtil.js";
    }
}
```

Zusätzlich muss diese Klasse in der Konfigurationsdatei „plugin.xml“ unter der Extension „org.eclipse.rap.ui.resource“ als „resource“ eingetragen werden, damit sie vom Laufzeitsystem geladen wird.

Als Nächstes wird auf dem Java Server ein Widget benötigt, welches auf dieses Ereignis reagieren kann. Dies wird durch ein „ClickLabel“, abgeleitet von „Label“, im Package „org.eclipse.rap.widgets“ realisiert, welches mit Hilfe der Methoden „addMouseListener“ und „removeMouseListener“ ein Interface „MouseListener“ als Parameter erhält, das für das Ereignishandling vorgesehen sind:

```
public class ClickLabel extends Label {

    public ClickLabel( final Composite parent, final int style ) {
        super( parent, style );
    }

    public void addMouseListener( final MouseListener listener ) {
        MouseEvent.addListener( this, listener );
    }

    public void removeMouseListener( final MouseListener listener ) {
        MouseEvent.removeListener( this, listener );
    }
}
```

Das Interface „MouseListener“ hält drei Methoden bereit, die vom Anwendungsentwickler zur weiteren Verarbeitung verwendet werden können. Die Methoden sind „mouseDoubleClick“ (doppelter Mausklick), „mouseDown“ (Maustaste wurde gedrückt) und „mouseUp“ (Maustaste wurde losgelassen):

```
public interface MouseListener extends EventListener {

    //Sent when a mouse button is pressed twice within the (operating
    //system specified) double click period.
    public void mouseDoubleClick( MouseEvent e );

    //Sent when a mouse button is pressed.
    public void mouseDown( MouseEvent e );

    //Sent when a mouse button is released.
    public void mouseUp( MouseEvent e );
}
```

Im nächsten Schritt werden die beiden Welten Java und JavaScript miteinander verbunden. Dazu wird dem RWT Widget ClickLabel der Lifecycle Adapter (LCA) „ClickLabelLCA“ im Package „org.eclipse.rap.internal.widgets.clicklabelkit“ zugewiesen, der für die Generierung des JavaScript zuständig ist. Dort wird die Anfrage an die MouseLCAUtil weiter delegiert. Zur Veranschaulichung wird nur auf die Methoden „addListener(...)“, „processEvent(...)“ und „createMouseEvent(...)“ näher eingegangen:

```

public class MouseLCAUtil {

    /*
     * Add a Double Click, Mouse Down and Mouse Up Event to the widget
     * The parameter pointer means, if the mouse graphic should be
     * changed to a pointer graphic (as links have)
     */
    public static void addListener( final Widget widget , boolean pointer) {

        try {
            JSWriter writer = JSWriter.getWriterFor( widget );
            writer.addListener("dbclick",
                "org.eclipse.rap.qx.MouseEventUtil.mouseDoubleClick");

            [some code...]

            if (pointer)
                writer.set("cursor", "pointer");
        }
        catch ( IOException ex ) {
            ex.printStackTrace();
        }
    }

    /*
     * In the lifecycle (renderChanges) data can be set once more
     */
    public static void updateListener(final Widget widget, boolean pointer)
    {[some code...]}

    //Removes all listeners
    public static void removeListener() {
        [some code...]
    }

    /*
     * This method processes the mouse click, which was sent from the
     javascript
     */
    public static void processEvent( final Widget widget ) {

        try {
            String eventId =
                "org.eclipse.rap.qx.MouseEventUtil.mouseDoubleClick";
            if( WidgetLCAUtil.wasEventSent( widget, eventId ) ) {
                createMouseEvent( widget, SWTConstants.MouseDoubleClick );
            }
            [some code...]
        }
        catch (Exception ex) {
            ex.printStackTrace();
        }
    }

    private static void createMouseEvent( final Widget widget,
        final int type ) {

```

```

MouseSelectionInfo mouse =
    new MouseSelectionInfo( (Control )widget,
        RequestUtil.getIntegerParameter( widget, "page.x" ),
        RequestUtil.getIntegerParameter( widget, "page.y" ));

    int x = mouse.getWidgetRelativeX();
    int y = mouse.getWidgetRelativeY();

    int screenX = RequestUtil.getIntegerParameter( widget,
                                                    "screen.x");
    int screenY = RequestUtil.getIntegerParameter( widget,
                                                    "screen.y");

    int pageX = RequestUtil.getIntegerParameter( widget, "page.x");
    int pageY = RequestUtil.getIntegerParameter( widget, "page.y");

String button = RequestUtil.getParameter( widget, "button");

MouseEvent event;
    event = new MouseEvent( widget,
                            null,
                            type,
                            button,
                            new Point(x, y),
                            new Point(pageX, pageY),
                            new Point(screenX, screenY));

    event.processEvent();
}

```

Wenn einem ClickLabel Widget mit „addMouseListener(...)“ ein Mausereignismelder zugewiesen wird, ist die Methode „addListener(...)“ der „MouseLCAUtil“ für das Schreiben des äquivalenten JavaScript Objekts zuständig. Dazu wird der „JSWriter“ für das Qooxdoo Widget instanziiert, der die statische JavaScript Methode dem Ereignis „dbclick“ zuweist, damit das JavaScript Objekt auf Doppelklicks hören kann. Mit dem Setzen der Werte „cursor“ bzw. „pointer“ mutiert der Mauszeiger zu einem Fingerzeiger. Die Methode „processEvent(...)“ erhält die ausgelösten Anfragen der statischen JavaScript Methode, die über die selbst definierte Ereignis ID identifiziert wird. Bei erfolgreicher Identifikation wird ein serverseitiges Mausereignis konstruiert. Dazu dient die private Methode „createMouseEvent(...)“, die die einzelnen Parameter der Anfrage, wie z. B. „page.x“ oder „page.y“, aus der Klasse „RequestUtil“, die die HTTP Anfrage analysiert, mitgeteilt bekommt. Am Ende der Methode wird ein Objekt „MouseEvent“ erstellt, welches mit den erhaltenen Parametern initialisiert und mit „event.processEvent“ ausgelöst wird. Eine ClickLabel, welches die Methode „mouseDoubleClick(...)“ implementiert, kann im übergebenen event Objekt die Mauskoordinaten über „event.x“ bzw. „event.y“ abfragen.

Die gesendeten Koordinaten der JavaScript Methode sind absolut, d. h. sie werden von der linken, oberen Bildschirmcke aus berechnet. In diesem Anwendungsfall sollen sie jedoch relativ zum Widget ClickLabel berechnet werden. Dazu bedarf es einer serverseitigen Transformation, die alle Eltern Widgets, wie z. B. einem Fenster (Shell) oder Composite (Container), iterieren muss. Dabei ergibt sich eine Ungenauigkeit von 3 Pixel, die jedoch völlig ausreichend ist. Das nächste Code Snippet wird mit den enthaltenen Kommentaren erklärt. Der entscheidende Algorithmus befindet sich in der Methode „computeRelativeMousePosition(...)“ und es bleibt dem Leser überlassen diesen zu verstehen.

```
public class MouseSelectionInfo {

    private Control control;
    private Point clientAbsolute, clientRelative, screen;

    public MouseSelectionInfo(Control control, int clientX, int clientY){
        this(control, clientX, clientY, 0, 0);
    }

    public MouseSelectionInfo(Control control, int clientX, int clientY,
                               int screenX, int screenY) {
        this.control = control;
        this.clientAbsolute = new Point(clientX, clientY);
        this.clientRelative =
            computeRelativeMousePosition(clientAbsolute);
        this.screen = new Point(screenX, screenY);
    }

    /*
     * The mouse coordinates are received absolute from the Java Script.
     * If you want to get the relative (inner) coordinates of the
     * control, the bounds of the control and those of its parents are
     * substracted. Further, if the window has Scrollbars
     * (ScrolledComposite), the bounds
     * of the control are distorted, because they are calculated by their
     * visibilty and not by logic. This means, when you scroll
     * horizontally or vertically to the Control "c" at the position "p",
     * and the scrollbar has the value "v", the bounds of c are "p-v" and
     * not "p". Therefore these values have to added. Finally, the method
     * is stopped, when the Control is a Shell, because its bounds are
     * given relative to the Display.
     */
    private Point computeRelativeMousePosition(Point absolute) {

        Point relative = new Point(absolute.x, absolute.y);

        Control currentControl = this.control;
        do {
            Rectangle bounds = currentControl.getBounds();
            relative.x -= bounds.x;
            relative.y -= bounds.y;

            //When Scrollbars, the value of the scrollbars has to be added
            if (currentControl instanceof ScrolledComposite) {
                ScrolledComposite sc =
                    (ScrolledComposite)currentControl;
            }
        } while (currentControl instanceof ScrolledComposite);
    }
}
```

```

        ScrollBar hb = sc.getHorizontalBar();
        ScrollBar vb = sc.getVerticalBar();
        relative.x += hb.getSelection();
        relative.y += vb.getSelection();
    }

    else {
        //Shell is last Control to be calculated
        if (currentControl instanceof Shell)
            break;
    }
    //Otherwise the next parent is processed
    currentControl = currentControl.getParent();
} while( currentControl != null );

//Unprecise calculation due to the script or getBounds()-Methods
//What the hell ever, fix it statically
relative.x -= 3;
relative.y -= 3;

return relative;
}
[some getter/setter methods...]
}

```

Der Mausereignismelder wurde nur für den Milestone 3 Release implementiert. Der folgende Milestone 5 Release beinhaltet eine Änderung in der Syntax, die den Mouse Listener inkompatibel machte. Ein Aktualisieren der Erweiterung war aber aufgrund von geänderten Anforderungen nicht mehr nötig, hätte aber keinen allzu großen Aufwand bedeutet.

Einiges wurde mit diesem Projekt erreicht und viele Fragen bleiben immer noch offen. Die wichtigsten Erkenntnisse dieser Arbeit werden im folgenden Schlusswort zusammengefasst, dass darüber hinaus einen Ausblick in die Zukunft gewährt.

## 5 Schlusswort

Wer in der Internetbranche erfolgreich sein will, braucht vor allem eines: originelle und innovative Ideen. Die Fähigkeit fundierte Konzepte dafür entwerfen zu können, schrieben sich viele Dot-Com Firmen selbst zu. Doch viele davon gingen unter, weil die nötige Fachlichkeit einfach fehlte. Fundierte Kenntnisse im Entwickeln von aktuellen Webtechnologien ist zwar eine zwingende Voraussetzung, jedoch garantiert dies allein auch nicht ein Bestehen in der schnelllebigen Welt des Internets. Es ist ferner Spezialwissen über Domänen von Nöten, um sinnvolle Systeme zweck- und zielorientiert zu konstruieren. Es reicht nicht nur aus, das Know-how zu besitzen, sondern auch das Know-why muss vorhanden sein. Ein höchst ausgeklügeltes CMS (Content Management System) mit allerlei Funktionalitäten und einem hohen Grad an Benutzerfreundlichkeit hat keine Chance, wenn die Zielgruppe lieber ihr Wissen in Foren oder Wikis stellt.

Softwarehäuser müssen ihren Kunden über ihre Produkte hinaus immer mehr Dienstleistungen bieten, um sie bei Laune zu halten. Denn eine Fragestellung wird sich vielen Produktmanagern in Zukunft immer öfter stellen: Wieso sollte ein Kunde ein teures Softwarepaket kaufen, wenn bereits gute Lösungen gratis zum Download verfügbar sind oder sofort als Service kostenlos benutzt und getestet werden können? Als polarisierende Beispiele können Google und Microsoft genannt werden, die sich nicht nur in der Entstehungsgeschichte, Philosophie, sondern auch im Geschäftsmodell unterscheiden. Die Großen müssen nur darauf achten, dass sie nicht zu einer Randerscheinung abdriften. Spezialisierungen in bevorzugten Bereichen sind notwendig und es wird in Zukunft nicht mehr möglich sein „alles für alle“ zu machen, wie es Microsoft gerne hätte. Schlagwörter wie „freie Services“ und „kostenloser Download“ sollten dennoch mit Vorsicht genossen werden. Hinter firmengetriebenen Projekten versteckt sich oft eine clevere Werbestrategie, die potentielle Kunden über eine Community an das Unternehmen zu binden versucht. Neben der kostenlosen Version existiert meist eine kommerzielle Ausgabe, die für einen professionellen Einsatz nützliche Funktionen und Support beinhaltet. Eine Aktualisierung ist meist problemlos und liegt auch im Sinne des Marktes. Der Vorteil zur Vergangenheit besteht darin, dass man die Produkte selbst austesten kann und sich nicht mehr auf die Beschreibungen des Herstellers verlassen muss. Ein gutes Open Source Projekt ist demnach die beste Werbung für ein Unternehmen. In der kommerziellen Version bekommt man mindestens das Gleiche, aber meist mit besserer Performanz, Stabilität und benutzerfreundlicheren Oberflächen.



Den genannten Entwicklungen muss sich auch die Firma BOC gemeinsam mit ihrer ADO-Produktfamilie stellen und sich überlegen, wie sie auf die zunehmende Serviceorientierung und auf die offensichtlichen Veränderungen in der Geschäftswelt antwortet. Eine erste Reaktion darauf ist ADOWeb, welches zwar konzeptionell gut positioniert ist, aber noch Mängel und „Kinderkrankheiten“ in der Umsetzung hat. Die Szenarien in Kapitel 2.3.3 beschreiben zwar nur einen kleinen Teil der Möglichkeiten, sie geben jedoch schon einen Ausblick darauf, wie eine ADOWeb Community Plattform gestaltet werden könnte. Da die BOC mit 140 Mitarbeitern die Größe eines Mittelbetriebs besitzt, sind die technischen und organisatorischen Möglichkeiten begrenzt. Die Forschungsgruppe finanziert sich hauptsächlich aus internen, nationalen und EU-Projekten. Mit dem Aufkommen des Community- und Web 2.0-Gedanken erschließen sich neue Möglichkeiten und Dimensionen der Zusammenarbeit. Einerseits kann man sich bei den Grundlagen auf Expertengemeinden, wie z. B. in der Softwareentwicklung auf Eclipse, verlassen, um andererseits die eigene Communitybildung voranzutreiben, deren Benutzerbeiträge zu ADOWeb Fortschritten führen soll. Blogs, Wikis, Newsgroups, Foren und andere soziale Umgebungen bieten hierzu reichliche Möglichkeiten und des Weiteren wäre überlegenswert, ob nicht Teile des Projekts als Service bzw. als downloadbares Archiv freigegeben werden sollen. Vorstellbar wäre die Freigabe des Datenmodells in Form der AdoScript Objects Bibliothek. Dies würde eine Java API für die Modellierung bereitstellen, um programmatisch Modelle bzw. Objekte anzulegen, zu editieren, Information abzufragen und löschen zu können. Die grafische Darstellung könnte in einem einfachen JavaScript Canvas erfolgen, der die geschriebene Applikation in einer Test- bzw. Visualisierungsumgebung grafisch interpretiert, um eine spielerische Einarbeitung in die API zu ermöglichen. Das Ablegen der Modelle könnte über eine lokale Datenbank oder einen gehosteten Service, der die Daten auf einem zentralen Server verwaltet, erfolgen. Kontributionen könnten über ein CVS, welches ein Software Repository für Zwecke der Teamarbeit bereit stellt, eingeholt werden. Der Betrieb müsste von firmeninternen Mitarbeitern gepflegt werden, wie es z. B. bei der RAP Newsgroup der Fall ist. Eine grafische Benutzeroberfläche für den Modell Editor, die Kennzahlenauswertungen mit ADOScore oder konfigurierbare PROMOTE Szenarien wären dann in kommerziellen Versionen erhältlich. Die Qualität der API ist entscheidend, d. h. sie sollte erstens portabel, erweiterbar und einfach sein, und zweitens sollten geschriebene Open Source Applikationen problemlos auf die kommerziellen Versionen übertragbar sein. Den Ansporn zum Schreiben von Kontributionen sollten einige funktionierende Demos liefern, die das Potential der BOC Produkte andeuten.

Rich Internet Applikationen, Leichtgewichtigkeit und kollektive Intelligenz gehören zu den Schlagwörtern, die in den nächsten Jahren durch die Medien geistern werden. Wie es sich langfristig entwickeln könnte, wurde bereits im Abschnitt 2.4 skizziert. Das Web bietet durchaus die Infrastruktur und grundlegenden Funktionalitäten, um die virtuelle Realität beherbergen zu können. Benutzerbeiträge, Partizipation und Netzwerke sind die Grundlagen, um die Wirklichkeit in eine digitale Form zu gießen. Es kann heute nur über die Technologien spekuliert werden, die nötig sind, um auf ein virtuelles Seminar von Tim O'Reilly zu fahren, die in einem gemieteten, künstlichen Konferenzraum stattfinden, welche ein wunderbares Panorama auf eine Meeresküste bietet. Aber wer meint dies sei Zukunftsmusik, muss sich nur mal die Gegenwart anschauen und wird das Gleiche sehen: virtuelle Umgebungen, in der sich Menschen treffen, kommunizieren und austauschen. Die virtuelle Realität hat schon begonnen und wir sind mittendrin, statt nur dabei.

Konzepte wie das Web 2.0 oder Domänenmodellierungen mit ADOweb werden die Choreographie für die Bühne bereiten, auf der sich Programmierer und Techniker bewegen werden. Die Kunst besteht darin abzuwägen, welchen Richtlinien man vertrauen kann und welche man lieber verwerfen sollte. Eine theoretische Analyse für sich wird nicht ausreichen und ein engagiertes Projekt wird nicht über die Tatsache hinwegkommen, dass Bibliotheken und Frameworks eingesetzt werden müssen. Die neue Entwicklung von Komponenten wird in einem Projekt oft als die erste Option angesehen, und man tut wahrscheinlich nur dem eigenen Ego einen Gefallen, wenn das Rad neu erfunden wird. Es ist eines der Zwecke dieser Diplomarbeit zu zeigen, dass eine ausführliche Recherche verfügbarer Frameworks notwendig ist. Im Fall der Literaturdatenbank hat das RAP Framework zu großen Sprüngen verholfen und eine Ajax Applikation kreiert, die zwar noch etwas instabil und inperformant ist, jedoch an Aussehen und Funktionalität beeindruckt. Das Konzept hat die erste Prüfung überstanden. Aufgrund der großen Anzahl an Ajax Frameworks und dem heißen Konkurrenzkampf ist es nahezu auszuschließen, dass die Technologie in nächster Zeit eingestellt wird. Es kann nur passieren, dass etwas Besseres erfunden wird, welches JavaScript überflüssig macht. Browser Plugins, wie z. B. Flash oder Java Applets, scheinen zurzeit noch ernsthafte Mitstreiter zu sein, während das browser-basierte XUL schon seit längerem auf einen größeren Durchbruch wartet. Möglich ist auch, dass ein neuer Client

erfunden wird, der den Browser in den Schatten stellt, wer weiß. In der Geschichte der Informatik ist es aber bisher selten so gekommen, wie man zuvor vermutet hätte.<sup>145</sup>

Die Literaturdatenbank zeigt Dimensionen des Browsers, die man sich vor einigen Jahren nicht hätte vorstellen können. Wie es bei Open Source Projekten üblich ist, war die karge Dokumentation des RAP Frameworks ein großes Defizit. Als Ersatz diente nur eine Newsgroup, die aber von kompetenten RAP Insidern gewartet wird, wie im Abschnitt 4.5 anhand der Implementierung eines Mausereignismelders geschildert wurde. Die Erweiterung der Plattform war harte Arbeit, gewährte aber interessante Einblicke in die RAP Logik. Eine Nebenerscheinung der Ajax Literaturdatenbank ist AdoScript Objects. Diese Bibliothek wurde parallel zum GUI entwickelt, um die eigentlichen Applikationsentwicklung zu beschleunigen. Zu Anfang war diese Doppelbelastung eher ein Hindernis als eine Hilfe, aber als die AdoScript Objects Datenmodell Bibliothek fertig war, konnten Funktionalitäten im GUI schnell und einfach umgesetzt werden. In diesem Projekt wurde sie als Client verwirklicht, jedoch empfiehlt sich im Hinblick auf die Integration in das ADOweb Service Framework eine Transformation zu einem Service. Der Einsatz von AdoScript Objects war mit einigen Adaptionen auch in anderen Projekten erfolgreich, die die Entwicklung um einiges vereinfachte und beschleunigte. Alles in allem kann die Auswahl des RAP Frameworks, die Implementierung des GUI und der Geschäftslogik der Literaturdatenbank als sehr zufrieden stellend eingestuft werden. Es ergaben sich natürlich auch einige Probleme, wie z. B. das Fehlen einer persistenten Schicht, die aber in zukünftigen Projekten bewältigt werden sollen. Es wurde sehr viel GUI Funktionalität entwickelt. Das Gesamtfazit dieser Arbeit kann als durchwegs positiv und zufrieden stellend bewertet werden. Der Wert des Geschriebenen und Gemachten wird sich in der Zukunft zeigen, inwieweit es auch für andere Projekte, die sich in Richtung Ajax, Web 2.0 oder Modellierung bewegen, nützlich sein wird.

---

<sup>145</sup> Der Gründer von IBM, Thomas J. Watson, soll 1943 behauptet haben.: „I think there is a world market for maybe five computers“

## 6 Literaturverzeichnis

**1 & 1 Internet AG:** „*Qooxdoo*“, URL: <http://qooxdoo.org/> (abgerufen am 18. April 2007)

**ACM:** „*Association for Computing Machinery*“, URL: <http://www.acm.org/> (abgerufen, am 6. September 2007)

**Adobe:** „*Photoshop CS3 Edition*“,  
URL: <http://www.adobe.com/products/photoshop/index.html>  
(abgerufen am 5. Juli 2007)

**Alby, Tom:** „*Web 2.0*“, Konzepte, Anwendungen, Technologien, 2.. Auflage, Hanser Verlag München, 2007

**Almaer, Dion et al:** „*Ajaxian.com 2006 Survey Results*“ , In: Ajaxian.com, URL: <http://ajaxian.com/archives/ajaxiancom-2006-survey-results> (abgerufen am 1. Juli 2007)

**Alonso, Gustavo et al:** „*Web Services*“, Springer-Verlag Berlin Heidelberg, Berlin/Deutschland, 2004

**Appel, Frank u. Krause, Jochen:** „*Developing with the Rich Ajax Platform*“. White Paper from Innoopract Informationssysteme GmbH, 2007

**AsIsKnown:** „*A semantic-based knowledge flow system for the European home textiles industry*“, URL: <http://www.asisknown.org> (abgerufen am 8. Mai 2007)

**Balzert, Helmut:** „*Lehrbuch der der Software-Technik.*“, Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung Spektrum, 1998

**Basic Thinking Blog:** „*Unterschied Chat, Forum, Newsgroup, Blog*“, URL: <http://www.basicthinking.de/blog/2005/12/21/unterschied-forum-chatnewsgroup-blog/>, verfasst am 21.12.2005, (abgerufen am 29. März 2007)

**Battson, Tobias:** „*Ajax Write 0.9 – Word Editor Online*“,  
URL: <http://www.ajaxwrite.de/> (abgerufen am 29. März 2007)

**Berners-Lee, Tim:** „*Transcript of Tim Berners-Lee's talk to the LCS 35th Anniversary celebrations*“, Cambridge Massachusetts, April 1999,  
URL: <http://www.w3.org/1999/04/13-tbl.html> (abgerufen, am 6. September 2007)

**BlinkBits.com:** „*Google Wikipedia RSS Feed*“, In: www.blinkbits.com, verfasst am 28. März 2007, URL: [http://www.blinkbits.com/de\\_wikifeeds\\_rss/Google](http://www.blinkbits.com/de_wikifeeds_rss/Google) (abgerufen am 29. März 2007)

**BOC:** „*Geschäftsprozess- und Unternehmensmodellierung*“, URL: <http://boc-eu.com> (abgerufen am 29. März 2007)

**BOC:** „Prozessorientierte Anwendungsentwicklung mit ADONIS“, BOC White Paper. URL: [http://www.boc-group.com/documents/products/adonis\\_whitepaper\\_de.pdf](http://www.boc-group.com/documents/products/adonis_whitepaper_de.pdf) (abgerufen am 29. März 2007)

**Bovet, Daniel P. und Cesati, Marco:** „*Understanding the Linux Kernel*“, 3. Auflage, O'Reilly, USA, 2005

**Bronner Online AG:** „*Standard.at*“, Wien, 2007, URL: <http://standard.at> (abgerufen am 8. Mai 2007)

**Bruckmeier, Jürgen:** „*Wenig Lust auf Vista*“, In: *Tomorrow*, Ausgabe Mai 2007, Deutschland

**Charland, Andre:** „*Will AJAX Replace the Desktop?*“, In: *Developer.com*, URL: <http://www.developer.com/java/web/article.php/3574116> (abgerufen am 27. September 2007)

**Chou, Wu et al:** „*Web Services for Service-Oriented Communication*“, In: *IEEE*, New Jersey/USA, 2006

**Cloudgarden:** „*Jigloo SWT/SWING Gui Builder for Eclipse*“, URL: <http://www.cloudgarden.com/jigloo> (abgerufen am 8. Mai 2007)

**CreamTec:** „*AjaxSwing*“, URL: <http://www.creamtec.com/products/ajaxswing/demos.html> (abgerufen am 30. Dezember 2007)

**Deutsche Bank Research:** „*Dot-Com Crash: Talsohle in Deutschland durchschritten*“, Frankfurt am Main/Deutschland, 15. August 2003, URL: [http://www.dbresearch.com/PROD/DBR\\_INTERNET\\_EN-PROD/PROD00000000000062570.pdf](http://www.dbresearch.com/PROD/DBR_INTERNET_EN-PROD/PROD00000000000062570.pdf) (abgerufen am 18. Februar 2008)

**del.icio.us corporate:** „*del.icio.us – social bookmarking*“, URL: <http://del.icio.us/> (abgerufen am 30. Dezember 2007)

**Denny Carl:** „*Praxiswissen Ajax*“ In: O'Reilly, 1. Auflage, Köln/Deutschland, 2006

**DKE – Department of Knowledge and Business Engineering:** „*Eine Symbiose zwischen universitärer Forschung und industrieller Anwendung*“, Wien/Österreich, 2007. URL: <http://www.dke.univie.ac.at/ueberuns/index.php?position=ueber%20uns> (abgerufen, am 29. August 2007)

**Dojo Foundation:** „*Dojo – The JavaScript Toolkit*“, URL: <http://dojotoolkit.org/> (abgerufen am 20. Juli 2007)

**Eclipse:** „*An open development platform*“, URL: <http://www.eclipse.org> (abgerufen am 8. Mai 2007)

**Eclipse Foundation:** „*ATF (AJAX Toolkit Framework)*“, URL: <http://www.eclipse.org/atf/> (abgerufen am 1. Oktober 2007)

**Eclipse Foundation:** „*RAP – Rich Ajax Platform*“, URL: <http://www.eclipse.org/rap> (abgerufen am 1. Oktober 2007)

**EDGAR Online:** „NASDAQ Composite IXIC“, URL: <http://charting.nasdaq.com/ext/charts.dll?2-1-14-0-0-354-03NA000000COMP-&SF:40|41|42-HC=0> (abgerufen am 18. Februar 2008)

**Encyclopædia Britannica Online:** „Collaboration“, URL: <http://www.britannica.com/dictionary?book=Dictionary&va=Collaboration&vao=Entry+Word&rh=&rho=Rhyme&fl=&sl=&et=&dt=&df=&dfo=Defining+Text> (abgerufen am 18. Februar 2008)

**Eichner, Hannes Dipl.-Ing:** „ADOWeb Architektur.“ Internes Dokument, Wien/Österreich, Juli 2007

**Fartell, Jason und Nezek, George S.:** „Rich Internet Applications – The Next Stage of Web Application Development“. Proceedings of the ITI 2007 29th Int. Conf. on Information Technology Interfaces, June 25-28, 2007 Cavtat, Croatia. Grand Valley State University, Allendale/MI/USA, 2007

**Fiveruns:** „Rico“, URL: <http://openrico.org> (abgerufen, am 1. August 2007)

**Flanagan, David:** „JavaScript: The Definitive Guide“, In: O'Reilly & Associates, USA, 2001

**Frentz, Clements:** „Die Chronik einer Kapitalvernichtung“, In: manager-magazin.de, Hamburg 1.6.2003, URL: <http://www.manager-magazin.de/geld/artikel/0,2828,186368-2,00.html> (abgerufen, am 19. Februar 2008)

**Freitag, Pete:** „Ajax Tutorial with Prototype“, New York/USA, Dezember 2005, URL: <http://www.petefreitag.com/item/515.cfm> (abgerufen am 1. Oktober 2007)

**Garret, Jesse:** „Ajax: A New Approach to Web Applications“, Adaptive Paths, San Francisco/USA, Februar 2005

**Getahead:** „Direct Web Remoting (DWR)“, URL: <http://www.getahead.ltd.uk/dwr/> (abgerufen am 20. Juli 2007)

**Gissing, Bernhard u. Tochtermann, Klaus:** „Corporate Web 2.0“, Shaker Verlag Aachen, 2007

**Google:** „Google AdSense“, URL: <https://www.google.com/adsense/login/de/index.html> (abgerufen am 2. März 2008)

**Google:** „GWT (Google Web Toolkit)“, URL: <http://code.google.com/webtoolkit/> (abgerufen am 20. Juli 2007)

**Kappel, Gerti et al:** „Modeling Customizable Web Applications“ In: IEEE 01, Department of Information System (IFS), University of Linz, Linz/Österreich, 2001

**German, Kent:** „Top ten dot-com flops“, In: CNET.COM, URL: [http://www.cnet.com/4520-11136\\_1-6278387-1.html?tag=cnetfd.sd](http://www.cnet.com/4520-11136_1-6278387-1.html?tag=cnetfd.sd) (abgerufen am 19. Februar 2008)

**Google:** „*Google Maps*“, URL: <http://maps.google.de/> (abgerufen am 29. März 2007)

**Google Scholar:** „*Scholar.google.com*“, URL: <http://scholar.google.at/> (abgerufen, am 20. Mai 2007)

**Hassan, Ahmed E und Holt, Richard C.:** „*Architecture Recovery of Web Applications*“, In: ICSE 02 19.-25. Mai, Orlando/USA, Mai 2002

**Hauswirth, Manfred und Dustdar, Schahram:** „*Peer-to-Peer: Grundlagen und Architektur*“. URL: <http://www.infosys.tuwien.ac.at/Staff/sd/papers/DBS-P2P.pdf> (abgerufen, am 20. Mai 2007)

**Hawlik, Wolfgang:** „*Symbiose zwischen Forschung & Anwendung*“. In: OCG 2/2001, Wien/Österreich, Juni 2001

**Herz, Andreas:** „*Open-jACOB*“, URL: <http://www.openjacob.org/> (abgerufen, am 20. Mai 2007)

**Holz, Patrick:** „*Nutzung kollektiver Intelligenz*“. In: Two Zero, Stand: 18. Mai 2006, URL: [http://twozero.uni-koeln.de/content/e14/e30/index\\_ger.html](http://twozero.uni-koeln.de/content/e14/e30/index_ger.html) (abgerufen am 29. März 2007)

**IEEE:** „*The world's leading professional association for the advancement of technology*“, URL: <http://iee.org/portal/site> (abgerufen, am 20. Mai 2007)

**ISC - Internet Systems Consortium:** „*Number of Internet Hosts*“, URL: <http://www.isc.org/index.pl?/ops/ds/> (abgerufen, am 20. August 2007)

**ISO – International Standardization Organisation und IEC - International Engineering Consortium:** „*Information technology – Open Systems Interconnection – Basic Reference model: The Basic Model*“, 2. Auflage, Genf/Schweiz, 1994

**Isomorphic Software:** „*Smart Client – Ajax Gui System*“, URL: <http://www.smartclient.com> (abgerufen am 29. März 2007)

**Jazaheri, Prof. Medi:** „*Distributed Systems - Lecture 2: Communication*“. Vorlesungsfoliensatz des Instituts für Verteile Systeme an der TU Wien, Wien/Österreich, 2005

**JXTA(TM) Community Project:** „*JXTA*“. URL: <https://jxta.dev.java.net/> (abgerufen, am 20. Mai 2007)

**Karagiannis, Dimitris und Telesko R. (2000):** „*The EU-Project PROMOTE: A Process-oriented Approach for Knowledge Management*“, In: Proceedings of the Third International Conference on Practical Aspects of Knowledge Management (PAKM 2000), Oktober 30-31, 2000, Basel/Schweiz, 2000

**Krüger, Guido:** „*JAVA 2. Handbuch der Java-Programmierung*“, 2. Auflage, Addison Wesley Verlag, München/Deutschland, 2000

- Linden Lab:** „*Second Life*“, San Francisco/USA, 2003, URL: <http://secondlife.com> (abgerufen, am 1. Juli 2007)
- Linux Online:** „*What is Linux*“, URL: <http://www.linux.org/info/index.html> (abgerufen am 2 März 2008)
- Lui, Feng et al.:** „*Web Service for Distributed Communication Systems*“, In: IEEE, 2006
- Mahemoff, Michael:** „*Ajax Design Patterns*“, 1. Auflage, O'Reilly Verlag, San Francisco, USA, Juni 2006
- Mahemoff, Michael:** „*Ajax Patterns*“, URL: <http://ajaxpatterns.org> (abgerufen, am 1. Juli 2007)
- Meyen, Sebastian:** „*EclipseCon: Eclipse Rich Ajax Platform - ein Interview mit Jochen Krause*“, URL: <http://www.innoopract.com/de/news-events/news-und-presse-meldungen/article/eclipsecon-eclipse-rich-ajax-platform-ein-interview-mit-jochen-krause.html> (abgerufen, am 1. Juli 2007)
- Mintert, Stefan u. Leisegang, Christoph:** „*Ajax*“, 1. Auflage, dpunkt Verlag, Heidelberg/Deutschlang, 2007
- Microsoft:** „*MS Office Online*“, URL: <http://office.microsoft.com/de-de/default.aspx> (abgerufen am 5. Juli 2007)
- Microsoft:** „*ASP.NET*“, URL: <http://www.asp.net/default.aspx?tabindex=7&tabid=47> (abgerufen, am 10. August 2007)
- Napster LLC:** „*Free Napster*“, URL: <http://free.napster.com/> (abgerufen, am 10. August 2007)
- O'Reilly, Tim:** „*What is Web 2.0*“, URL: <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html> (abgerufen am 22. Mai 2007)
- Otte, Max Prof. Dr.:** „*GOOBAYHOO!*“, In: boerse.de (Expertenkolumnen), Stand: 17. März 2006, URL: <http://nachrichten.boerse.de/background.php3?text=experten&id=2022a804> (abgerufen am 29. März 2007)
- OpenJacob:** „*Building Databased Webapplications*“, URL: <http://www.openjacob.org> (abgerufen am 8. Mai 2007)
- OpenJacob:** „*Draw 2D*“, URL: <http://www.openjacob.org/draw2d.html> (abgerufen am 8. Mai 2007)
- Platon:** „*Phaidros (oder Von dem Schönen)*“, Übersetzung: F.D.E. Schleiermacher, 1826, URL: <http://www.textlog.de/34993.html> (abgerufen, am 5. August 2007)
- Postel, Jonathan:** „*Internet Protocol*“. In: RFC 791 (Requests for comments). Marina del Rey/USA, 1981. URL: <http://tools.ietf.org/html/rfc791> (abgerufen, am 20. August 2007)



**Powell, Thomas:** „AJAX is the future of Web app development“, In: Network World, 17. Juli 2006, URL: <http://www.networkworld.com/research/2006/071706-ajax-future.html> (abgerufen, am 20. August 2007)

**Pflüger, Jörg und Purgathofer, Peter:** „Vorlesung Informatik & Gesellschaft 1.“ Globalisierung vs. Netzkultur. Wien/Österreich, 2003

**Preciado, J. C.:** „Necessity of methodologies to model Rich Internet Applications“, In. IEEE 2005, Proceedings of the 2005 Seventh IEEE International Symposium on Web Site Evolution (WSE'05)

**Prototype Core Team:** „Prototype JavaScript Framework“, URL: <http://prototype.conio.net> (abgerufen, am 20. Juni 2007)

**Prototype Core Team:** „Who's using Prototype“, URL: <http://www.prototypejs.org/real-world> (abgerufen am 20. Juli 2007)

**RAP:** „Rich Ajax Platform“, URL: <http://www.eclipse.org/rap> (abgerufen am 8. Mai 2007)

**Rauschmayer, Axel:** „Ajax mit DWR und Dojo“, Deutschland, Dezember 2006, URL: <http://www.pst.ifi.lmu.d/~rauschma/articles/ajax-einf/index.html> (abgerufen am 1. Oktober 2007)

**Resig, John:** „jQuery,“ URL: <http://jquery.com/> (abgerufen am 20. Juli 2007)

**RP ONLINE GmbH:** „Google: Gewinn verdreifacht, Anleger enttäuscht“, In: Rheinische Post, Düsseldorf/Deutschland, 2007, URL: <http://www.rp-online.de/public/article/aktuelles/wirtschaft/news/402858> (abgerufen am 2. März 2008)

**Scheper, Ulrich Mag. (2007):** „ADONIS – Customizing“. Schulungsunterlagen des DKE der Universität Wien, Wien/Österreich, Mai 2007

**Sharman Networks Ltd:** „Kazaa“, URL: <http://www.kazaa.com/> (abgerufen am 20. Juli 2007)

**Snell, James:** „Call SOAP Web services with Ajax (Part 2: Extending the Web service client)“, URL: [http://www.ibm.com/developerworks/webservices/library/ws-wsajax2/index.html?S\\_TACT=105AGX04&S\\_CMP=EDU](http://www.ibm.com/developerworks/webservices/library/ws-wsajax2/index.html?S_TACT=105AGX04&S_CMP=EDU) (abgerufen am 10. Juli 2007)

**studiVZ Ltd.:** „studiVZ“, URL: <http://www.studivz.net/> (abgerufen am 19. Februar 2008)

**Sun Developer Network:** „Java Technology“, URL: <http://java.sun.com/> (abgerufen am 27. August 2007)

**Szogat, Martin et al.:** „Social Software“. In.: entwickler.press. Paderborn, 2006

**SWT:** „The Standard Widget Toolkit“, URL: <http://www.eclipse.org/swt> (abgerufen am 8. Mai 2007)

**Tanenbaum, Andrew und Van Steen, Martin:** „Verteilte Systeme.“ Pearson Studium, München/Deutschland, 2003

**The Linux Counter Project:** „*Estimating the number of Linux users*“, URL: <http://counter.li.org/estimates.php> (abgerufen am 2. März 2008)

**ThinWire LLC:** „*ThinWire*“, URL: <http://www.thinwire.com/> (abgerufen am 15. Juli 2007)

**TU und Universität Wien:** „*KFK Semantic Web.*“, URL: <http://www.dke.univie.ac.at/semanticweb> (abgerufen am 29. März 2007)

**TU Wien:** „*Studienpläne Informatik*“, Fakultät für Informatik  
URL: <http://www.informatik.tuwien.ac.at/studium/bachelor/studienplan-INF2006.pdf>  
(abgerufen, am 13. August 2007)

**Vise, David A.:** „*The Google Story*“, Bantam Dell Publishing Group, USA, 2005

**W3C:** „*Cascading Style Sheets*“, URL: <http://www.w3.org/Style/CSS/>  
(abgerufen am 26. August 2007)

**W3C:** „*Extensible Markup Language*“, URL: <http://www.w3.org/XML/>  
(abgerufen am 27. August 2007)

**W3C:** „*Extensible Stylesheet Language*“, URL: <http://www.w3.org/TR/xsl/>  
(abgerufen am 27. August 2007)

**W3C:** „*HTML 4.01 Specification*“, URL: <http://www.w3.org/TR/REC-html40/>  
(abgerufen am 27. August 2007)

**W3C:** „*Hyperlinks*“, URL: [http://www.w3.org/MarkUp/html-spec/html-spec\\_7.html](http://www.w3.org/MarkUp/html-spec/html-spec_7.html)  
(abgerufen am 27. August 2007)

**W3C:** „*Naming and Addressing*“, URL: <http://www.w3.org/Addressing/>  
(abgerufen am 27. August 2007)

**W3C:** „*OWL Web Ontology Language*“, 2004, URL: <http://www.w3.org/TR/owl-features/>  
(abgerufen am 27. August 2007)

**W3C:** „*RDF Resource Description Framework*“, 2004, URL: <http://www.w3.org/TR/rdf-syntax-grammar/> (abgerufen am 27. August 2007)

**W3C:** „*Scalable Vector Graphics (SVG)*“, URL: <http://www.w3.org/TR/SVG/>  
(abgerufen am 27. August 2007)

**W3C:** „*Simple Object Access Protocol (SOAP) 1.1*“, Mai 2000, URL: [http://www.w3.org/TR/2000/NOTE-SOAP-20000508/#\\_Toc478383487](http://www.w3.org/TR/2000/NOTE-SOAP-20000508/#_Toc478383487) (abgerufen am 15. Februar 2008)

**W3C:** „*World Wide Web Consortium*“, URL: <http://www.w3.org/>  
(abgerufen am 5. Juli 2007)

**W3C:** „*XHTML 1.1*“, URL: <http://www.w3.org/TR/xhtml11/>

(abgerufen am 27. August 2007)

**Wacker, Swen:** „*SELFHTML – MIME-Typen*“.

URL: <http://de.selfhtml.org/diverses/mimetypen.htm> (abgerufen am 17. August 2007)

**Wacker, Swen:** „*SELFHTML – JavaScript*“.

URL: <http://de.selfhtml.org/javascript/> (abgerufen am 27. August 2007)

**White Jared und Wilson, Max:** „*Xajax*“, URL: <http://www.xajaxproject.org/> (abgerufen am 20. August 2007)

**Wikimedia Foundation Inc.:** „*Wikipedia*“, USA, URL: <http://de.wikipedia.org/> (abgerufen am 20. August 2007)

**Wikimedia Foundation Inc.:** „*WYSIWYG*“, USA, URL:

<http://de.wikipedia.org/wiki/WYSIWYG> (abgerufen am 20. August 2007)

**Woitsch, Robert:** „*ADOWeb – Overall Architecture and Definition 1.0B*“, Internes Dokument, 1. Version, Wien/Österreich, 19. Oktober 2005

**Wollzelle GmbH:** „*Script.aculo.us*“, Wien/Österreich, 2007, URL: <http://script.aculo.us/> (abgerufen am 20. Juli 2007)

**Yahoo! UI:** „*The Yahoo! User Interface Library (YUI)*“, URL:

<http://developer.yahoo.com/yui/> (abgerufen am 20. Juli 2007)

**YouTube, LLC:** „*Broadcast Yourself*“ URL: <http://de.youtube.com/> (abgerufen am 10. Februar 2008)

**Zara, Olivier:** „*Managing Collective Intelligence, Toward a New Corporate Governance*“, M21 Editions, Paris, 2004, URL:

[http://www.axiopole.com/pdf/Managing\\_collective\\_intelligence.pdf](http://www.axiopole.com/pdf/Managing_collective_intelligence.pdf) (abgerufen am 15. Februar 2008)

**Zhao, Weiquan und Chen, Jian:** „*A Component Oriented Web Application Model*“, In: IEEE, University of South Australia, Adelaide/Australia, 1999