

KOS.content

02 | 2013

Ergebnisse der Untersuchungen des
Kompetenzzentrum Open Source der DHBW-Stuttgart

Sommer 2013
ergebnisband

INHALT BAND.1

Inhalt __

Entwicklung eines Konzeptes für eine Anwendung zur Bearbeitung von PDF-Dokumenten __ 1

Literaturrecherche: Webengineering - Modellgestützte Generierung von Oberflächen __ 119

Evaluation of Java-based TIFF processing approaches - a prototype for a German mid-size insurance company __ 43

A comparison of SAP HANA with Relational Databases __ 179

Marktanalyse clientseitiger OS Frameworks zum Bau browserbasierter Desktop-Anwendungen __ 69

Introduction to MongoDB __ 217

Das Kompetenzzentrum Open Source (KOS)

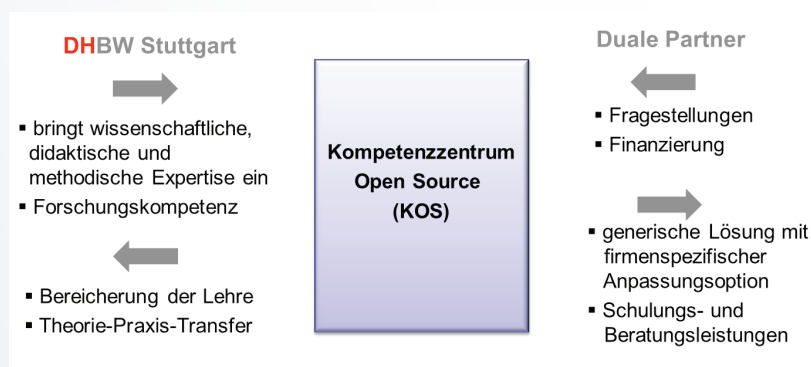
Ziel des Projektes

Das Projekt Kompetenzzentrum Open Source der DHBW Stuttgart wurde mit der Zielsetzung ins Leben gerufen, die Einsatzfelder für Open Source Software in Unternehmen zu identifizieren und durch den Einsatz quelloffener Produkte und deren kostengünstigen Einsatzmöglichkeiten Optimierungen in ausgewählten Geschäftsbereichen zu erzielen.

Dies bedeutet konkret, dass z.B. Open Source Software evaluiert wird, um Lizenzkosten zu reduzieren, bewertet wird, ob sie diverse Qualitätskriterien erfüllt und erfolgreich(er) und effizient(er) in Unternehmen genutzt werden kann. Das Ziel des Projektes ist es hierbei, allgemeingültige Lösungskonzepte für Problemstellungen zu erarbeiten, welche von den am Projekt beteiligten Unternehmen zu firmenspezifischen Lösungen weiterentwickelt werden können. Die beteiligten Unternehmen partizipieren so an den Ergebnissen des Projekts.

Zusammenarbeit mit den Dualen Partnern

Die Zusammenarbeit mit den Dualen Partnern gestaltet sich entlang deren Anforderungen und Bedürfnissen. Sie sind die Themengeber für betriebliche Fragestellungen, die im Rahmen des Projekts untersucht werden. Die DHBW steuert die wissenschaftliche, didaktische und methodische Expertise und Forschungskompetenz bei und untersucht die identifizierten Themenfelder.



Im Rahmen des Projektes steuert die DHBW Stuttgart die wissenschaftliche Expertise und Forschungskompetenz bei zur Bearbeitung der betrieblichen Fragestellungen der Dualen Partner. Es entstehen generische Lösungen, welche von den Partnern an Ihre Situation angepasst werden kann.

Im Rahmen der Arbeit entstehen (generische) Lösungen, an denen die Partner teilhaben können indem sie diese auf ihre spezifische Unternehmenssituation anpassen. Zudem fließen die Ergebnisse in die Arbeit der DHBW ein, sodass hier dem Anspruch an eine hohe Anwendungs- und Transferorientierung ganz im Sinne einer kooperativen Forschung Rechnung getragen wird.

An den Ergebnissen des Projekts partizipieren die Dualen Partner Allianz Deutschland AG, die Deutsche Rentenversicherung Baden-Württemberg und die HALLESCHE Krankenversicherung a.G.

Entwicklung eines Konzeptes für eine Java- Web-Anwendung zur Bearbeitung von PDF- Dokumenten

Schriftliche Ausarbeitung
im Rahmen der Lehrveranstaltung „Projekt des 6. Semesters“
Kompetenzzentrum Open Source (KOS)

Vorgelegt von

Melanie Thum, Fabian Stellmacher,
Fabian Flaig, Larissa Königs

am 02.07.2013

Fakultät Wirtschaft
Studiengang Wirtschaftsinformatik
WWI2010

Inhaltsverzeichnis

Abkürzungsverzeichnis	IV
Abbildungsverzeichnis.....	V
Tabellenverzeichnis.....	VI
1 Einleitung	1
2 Aufgabenstellung	2
2.1 Aktuelle Situation	2
2.2 Anforderungen	3
3 Definitionen	5
3.1 Open Source Software	5
3.2 Unterschiedliche Lizenzarten	6
3.3 PDF.....	8
4 Auswahl und Kurzvorstellung der ausgewählten Java Libraries	9
4.1 Apache PDFBox.....	9
4.2 icePDF	10
4.3 iText.....	11
4.4 PDF Clown.....	12
4.5 PDFjet.....	13
4.6 TCPDF	13
5 Bewertung und Auswahl des geeigneten Produktes.....	15
5.1 Vorstellung und Gewichtung des Kriterienkatalogs.....	15
5.2 Bewertung der Libraries anhand des Kriterienkatalogs.....	16
5.3 Auswahl einer geeigneten Java Library.....	19
6 Dokumentation der Implementierung des Prototypes	21
6.1 Aufbau der Webanwendung.....	21
6.2 Vorstellung der Systemvoraussetzungen	22
6.3 Benötigte Anpassungen der Libraries.....	23
6.4 Implementierung des Prototyps.....	25
6.4.1 Aufbau der Datenbank und Zugriff darauf	25

6.4.2	Erstellung der Schichten	27
6.4.3	Funktion „Schwärzen“	28
6.4.4	Funktion „Notizen einfügen“	29
6.4.5	Funktion „Stempeln“	29
6.4.6	Weitere Funktionen	30
6.4.7	Speichern der Änderungen in der Datenbank.....	31
7	Fazit	32
	Quellenverzeichnisse	34
	Literaturverzeichnis.....	34
	Verzeichnis der Internet- und Intranet-Quellen.....	34

Abkürzungsverzeichnis

AGPL	Affere General Public License
BSD	Berkeley Software Distribution-License
CPL	Common Public License
GPL	General Public License
GUI	Graphical User Interface
ID	Identifikator
IDE	Integrated Development Environment
IT	Informationstechnologie
JDK	Java Development Toolkit
JSP	Java Server Page
LGPL	Lesser General Public Licence
MPL	Mozilla Public License
OSI	Open Source Initiative
PC	Personal Computer
PDF	Portable Document Format
PHP	Urspr.: Personal Home Page Tools, heute: Hypertext Preprocessor)
SQL	Structured Query Language
TIF	Tagged Image File Format
XML	Extensible Markup Language

Abbildungsverzeichnis

Abbildung 1: Kriterienkatalog.....	16
Abbildung 2: Ansicht der index.jsp.....	21
Abbildung 3: Aufbau der Webanwendung	22
Abbildung 4: Benutzungsoberfläche von icePDF	24
Abbildung 5: Verwendetes Datenbankschema	26
Abbildung 6: Button zum Schwärzen	28
Abbildung 7: Button zum Einfügen von Notizen.....	29
Abbildung 8: Button zum Einfügen eines Stempels	29
Abbildung 9: Button zum Highlighten eines Textes.....	30
Abbildung 10: Button zum "Rückgängig machen"	30

Tabellenverzeichnis

Tabelle 1: Bewertung der Funktionalitäten	17
Tabelle 2: Zusammenfassung der Open Source Merkmale	18
Tabelle 3: Übersicht der eingesetzten Software	23

1 Einleitung

Trotz zunehmender Digitalisierung gibt es in vielen Unternehmen immer noch Unterlagen in Papierform. Vor allem bei Versicherungen ist dies häufig der Fall, da Unterschriften auf Dokumenten im Original angefordert werden. Aus Gründen der Einfachheit und der Archivierbarkeit wird aber versucht, diese auf ein Minimum zu reduzieren. Hierzu werden Unterlagen nach Eingang eingescannt und in einer digitalen Version weiterbearbeitet. So werden zum Beispiel Kommentare, Unterschriften oder auch Stempel digital zu den Dokumenten hinzugefügt. Der Kunde, in dessen Auftrag diese Arbeit verfasst wurde, verwendet bisher ein kostenpflichtiges Produkt, mit dem die Bearbeitung von gescannten Dokumenten möglich ist. Im Rahmen eines Projektes des Kompetenzzentrums Open Source soll nun eine Webapplikation zur Bearbeitung von PDF-Dokumenten erarbeitet und implementiert werden.

Zunächst einmal wird die Aufgabenstellung erläutert und durch sowohl allgemeine als auch technische Gegebenheiten ergänzt. Ebenso gehört zu diesem Kapitel die Erläuterung der verschiedenen Anforderungen, die an den Prototypen, der im Laufe dieses Projektes implementiert werden soll, gestellt werden. In Kapitel 3 folgen dann verschiedene Definitionen, die rund um das Thema Open Source Software, Lizenzarten und dem PDF-Dateiformate erläutert werden müssen. Kapitel 4 und 5 beschäftigen sich schließlich mit der eigentlichen Marktanalyse einiger ausgewählter Software Produkte. Hierzu werden die einzelnen Produkte in Kapitel 4 zunächst vorgestellt und daraufhin in Kapitel 5 bewertet. Hier wird auch die Wahl eines Produktes begründet, mit welchem schließlich der Prototyp implementiert wird. Die Umsetzung und Implementierung wird in dem darauffolgenden Kapitel beschrieben und abschließend das Ergebnis kritisch betrachtet. Im Fazit werden dann noch einmal die Ergebnisse dieser Arbeit zusammengefasst und ein kurzer Ausblick gegeben.

2 Aufgabenstellung

Im Folgenden wird das Thema dieser Arbeit, die Situation des Kunden und die daraus resultierenden Anforderungen näher erläutert.

2.1 Aktuelle Situation

Mit über 200.000 vollversicherten Kunden im Jahr 2012 muss eine große Krankenversicherung eine große Menge eingehender Dokumente bearbeiten, verwalten und archivieren.

Zur Bewältigung dieser Aufgaben wird ein sogenanntes Dokumentenmanagementsystem verwendet. Ein solches stellt die Verwaltung aller betrieblichen Dokumente mithilfe einer oder mehrerer Datenbanken dar. Zu diesem Zweck werden alle eingehenden Dokumente zunächst digitalisiert, bevor sie dann in elektronischer Form an die zuständigen Stellen zur weiteren Bearbeitung oder auch zur Archivierung weitergeleitet werden.

Das Dokumentenmanagementsystem des Kunden läuft auf einem Websphere Server. Der Zugriff durch die Nutzer – die Mitarbeiter des Unternehmens – erfolgt über eine Browseranwendung. Diese wird im Internet Explorer Version 9 eingesetzt und mithilfe von einer aktuellen Java Version umgesetzt.

Das Dokumentenmanagementsystem beinhaltet, dass alle eingehenden papiergebundenen Dokumente umgehend digitalisiert und erst in dieser digitalisierten Form – entweder als PDF oder als TIF Datei – an die zuständigen Stellen weitergeleitet und archiviert werden.

Mit dieser Lösung steht der Kunde allerdings vor dem Problem, dass die digitalisierten Dokumente zur weiteren Bearbeitung häufig noch gestempelt, geschwärzt oder auch kommentiert werden müssen, diese Änderungen aber nur als Layer über die jeweiligen Originaldokumente im Archiv gelegt werden dürfen, da die Originaldokumente erhalten bleiben sollen.

Die vorliegende Arbeit stellt eine mögliche Lösung dieser Problematik mithilfe von Open Source Produkten dar.

2.2 Anforderungen

Im Folgenden werden die vom Kunden genannten Anforderungen erläutert und daraus in Kapitel 5 ein Kriterienkatalog erstellt. In der Marktanalyse wird dann anhand dieses Katalogs geprüft, welche der zur Auswahl stehenden Open Source Bibliotheken die Anforderungen des Kunden am besten erfüllen und somit für den Prototyp in Frage kommen. Kriterien, welche durch keine Bibliothek erfüllt werden, müssen manuell von den Autoren dieser Arbeit entwickelt und ergänzt werden.

Zu den funktionalen Anforderungen des Kunden gehört, dass eine Webanwendung (Applet) entwickelt wird, die folgende Funktionen ermöglicht:

- **Stempeln** des Dokuments (z.B. Eingangs- oder Bearbeitungsstempel)
- **Schwärzen** von Wörtern, Passagen oder auch Bildern (wobei laut Angabe des Kunden so gut wie alle Dokumente reine Textdokumente sind)
- **Einfügen** von **Kommentaren** an beliebigen Stellen im Dokument
- **Extrahieren** von Textpassagen
- **Hervorheben** bzw. markieren von Wörtern und Textpassagen (sogenanntes Highlighting) als optionale Funktion
- Durch die Webanwendung getätigte Änderungen „**rückgängig machen**“ als weitere optionale Funktion

Desweiteren ist es der Wunsch des Kunden, die Originaldokumente nicht zu ändern. Daher besteht ein weiteres Kriterium darin, dass die Änderungen in **Schichten, sogenannten Layern** gespeichert werden müssen, die auf die entsprechenden Dokumente gelegt werden und dadurch nach Bedarf ein- und ausgeblendet werden können. Gespeichert werden sollen alle Änderungen und die dazugehörigen Schichten bevorzugt in einer relationalen Datenbank. Da das Augenmerk aber nicht auf diesem Teil der Webanwendung liegt, bleibt es von Seiten des Kunden offen, auf welches Datenbankmanagementsystem zurückgegriffen wird.

Aufgrund vorhandener Programmierkenntnisse in Java besteht ein weiterer Wunsch des Kunden darin, dass die Webanwendung in dieser Programmiersprache ge-

schrieben ist. Daher werden in der Marktanalyse zwar vorrangig Java-Bibliotheken betrachtet. Jedoch wird auch eine PHP-Bibliothek als Vergleich hinzugezogen.

Die im Text „fett“ hinterlegten Wörter stellen zusammen den sich aus den Kundenanforderungen ergebenden Kriterienkatalog dar, der in Kapitel 5 verwendet wird.

Dieser wird darüber hinaus in Kapitel 5 um weitere Kriterien der Autoren ergänzt, welche für die Auswahl der Produkte, ebenso wie für die Implementierung eines geeigneten Prototyps wichtig sind.

3 Definitionen

Im folgenden Kapitel werden die Begriffe Open Source Software und deren Nutzungsrechte, sowie der Begriff PDF genauer erläutert.

3.1 Open Source Software

Bei Open Source Software handelt es sich um ein "Konzept, nach dem Programme mit ihrem Quellcode ausgeliefert werden. Jeder darf den Quellcode einsehen und verändern. Die Open Source Initiative (OSI) definiert Kriterien, die Open Source Software erfüllen soll."¹

Die von der Open Source Initiative definierten Kriterien gliedern sich in 10 verschiedene Punkte, die genau beschreiben, unter welchen Bedingungen Open Source Software weiterverbreitet werden darf. Diese werden im Folgenden kurz aufgelistet und erläutert²:

1. Free Redistribution

Die Lizenz darf frei verteilt werden und darf niemanden darin beschränken. Ebenso dürfen keine Gebühren für die Verteilung verlangt werden.

2. Source Code

Die angebotene Software muss inklusive dem Quellcode erhältlich sein und verteilt werden. Ist dies nicht der Fall muss bekannt gemacht werden, von wo der Quellcode kostenfrei bezogen werden kann. Er muss in einer Form sein, mit der ein Programmierer weiterarbeiten kann. Der Code darf nicht absichtlich kompliziert und undurchsichtig aufgebaut sein.

3. Derived Works

Bereits abgeänderter Code darf unter den gleichen Bedingungen wie die Original-Software verbreitet werden.

4. Integrity of the Author's Source Code

¹ Lackes, R./ Siepermann, M. (o.J.)

² Vgl. Open Source Initiative (o.J.)

Die Lizenz darf die Distribution des Source Codes in einer modifizierten Form nur dann einschränken, wenn die Lizenz die Distribution von "patch files" erlaubt.

5. No Discrimination against Persons or Groups

Durch die Lizenz dürfen keine individuellen Personen oder Gruppen von Personen benachteiligt werden.

6. No Discrimination Against Fields of Endeavor

Die Lizenz darf niemanden davon abhalten, die Software in einem speziellen Anwendungsgebiet zu verwenden (z.B. in der Genforschung).

7. Distribution of License

Für jeden, für den die Software zugänglich ist, muss ebenso die Lizenz frei zugänglich sein.

8. License Must Not Be Specific to a Product

Wird ein Teil der Software extrahiert und weiter verteilt, so gilt für diesen Teil die gleiche Lizenz wie für die Ursprungssoftware.

9. License Must Not Restrict Other Software

Die Lizenz darf keine andere Software einschränken, die zusammen mit der lizenzierten Software verteilt wird. Das bedeutet zum Beispiel, dass andere Software ebenfalls Open Source sein muss.

10. License Must Be Technology-Neutral

Die Lizenz muss plattformunabhängig verfügbar sein.

3.2 Unterschiedliche Lizenzarten

Bei den Lizenzen für Open Source Software unterscheidet man allgemein „in Lizenzen ohne Copyleft-Effekt, Lizenzen mit strengem Copyleft-Effekt, Lizenzen mit beschränktem Copyleft-Effekt, Lizenzen mit Wahlmöglichkeiten und Lizenzen mit Sonderrechten“³. Ihre Einordnung wird allerdings als „umstritten“ gesehen.

Lizenzen ohne Copyleft-Effekt

³ ifrOSS (o.J.a)

Bei Lizenzen ohne Copyleft-Effekt, kann der Nutzer eigene Softwareteile zur Ursprungssoftware hinzufügen und seinen eignen Teil selbst lizensieren. Dabei kann es sich auch um eine andere Open Source Lizenz handeln, als bei der Ursprungslizenz. Der unveränderte Teil der Software unterliegt aber weiterhin der Ursprungslizenz. Eine viel verwendete Lizenzvariante dabei ist die Berkeley Software Distribution-License (BSD).

Lizenzen mit strengem Copyleft-Effekt

Bei dieser Lizenzvariante „muss der Nutzer alle Weiterentwicklungen und Änderungen ausnahmslos der Ursprungslizenz unterstellen“⁴. Beispiel dafür sind GNU General Public License (GPL) und die Common Public License (CPL). Die GNU General Public License ist wohl die am häufigsten verwendete Open Source Lizenz und findet Anwendung bei ca 60% der Open Source Produkte⁵. Ein Vertreter der Common Public License ist zum Beispiel Eclipse, welches als Java-Entwicklungsumgebung verwendet wird.

Lizenzen mit beschränktem Copyleft-Effekt

Hierbei dürfen die Open Source Software – abhängig von der jeweiligen individuellen Lizenzvereinbarung – und eigene Softwareteile, die unter einer anderen Softwarelizenz stehen, miteinander kombiniert werden. Vertreter dieser Lizenzvariante sind unter anderem Mozilla Public License (MPL) und GNU Lesser General Public License (LGPL). Insbesondere LGPL ist eine weniger beschränkte Form der GPL.

Lizenzen mit Wahlmöglichkeiten

Hierbei ist die Wahl der Lizenz abhängig von der Veränderung, die an der Open Source Software vorgenommen wurde. Lizenzen die hierbei Anwendung finden, sind Perl Artistic License und die Clarified Artistic License.

Lizenzen mit Sonderrechten

Bei Lizenzen mit Sonderrechten entwickeln Unternehmen eigene Lizenzen und behalten sich mit diesen gewisse Sonderrechte vor.

⁴ Keller-Stoltenhoff, E. (2008),

⁵ Vgl. ifrOSS (o.J.b)

3.3 PDF⁶

PDF ist ein Dokumentenformat, das von Adobe Systems erfunden wurde und steht für Portable Document Format. Das Besondere an diesem Dateiformat ist die Plattformunabhängigkeit. Die Intention dieser Entwicklung besteht darin, die automatische Konvertierung einiger Textbearbeitungsprogramme, wie z.B. Microsoft Word zu umgehen, wenn Dokumente verschickt und in einer anderen Version des Bearbeitungsprogrammes geöffnet werden. Dabei werden oft die durch den Ersteller des Dokuments vorgenommenen Formatierungen gelöscht oder angepasst. Das PDF-Format ermöglicht es, dass die optische Beschaffenheit des Dokuments nicht verändert wird und auf jeder Plattform das gleiche Layout hat.

Verwendet werden Dokumente im PDF-Format in allen möglichen Bereichen. Vor allem in E-Journals findet man häufig Artikel in diesem Format. Mit Hilfe eines PDF Bearbeitungsprogramms lassen sich Unterschriften unter offizielle Dokumente setzen. Auch die Suche nach bestimmten Textpassagen ist möglich. Eine weitere Funktion ist die direkte Navigation zu bestimmten Textabschnitten mit Hilfe eines Inhaltsverzeichnisses. Um Dokumente sichern zu können, können sie mit Passwörtern belegt werden, ohne die das Öffnen der Datei nicht möglich ist. Diese Funktion machen sich vor allem Regierungen zu Nutze wenn sie wichtige Dokumente verschicken.

⁶ Vgl. Adobe Systems (o.J.)

4 Auswahl und Kurzvorstellung der ausgewählten Java Libraries

Zur Erstellung des bereits erwähnten Prototyps wurde im Rahmen dieser Arbeit zunächst eine Onlinerecherche durchgeführt. Diese diente dem Zweck, verschiedene Open Source Produkte, die den Anforderungen des Kunden entsprechen oder nahe kommen und sich somit für den Prototyp eignen, ausfindig zu machen.

Die so getroffene Vorauswahl bestehend aus sechs Bibliotheken wird nachfolgend kurz vorgestellt. In Kapitel 5 erfolgt dann eine Bewertung dieser und darauf basierend wird eine finale Auswahl getroffen.

4.1 Apache PDFBox⁷

Apache PDFBox ist ein Open Source Produkt mit dem es möglich ist, PDF-Dokumente mit unterschiedlichen Funktionalitäten zu bearbeiten. Es ist eine Java-basierte Software und wurde unter der Apache License Version 2.0 veröffentlicht.

Zum einen ist es mit dieser Software möglich, existierende Dokumente zu bearbeiten. Zum anderen ist es aber ebenfalls möglich eigene PDFs aus Teilen andere Dokumente neu zu erstellen. Für die Bearbeitung bereits existierender PDFs stehen verschiedene Möglichkeiten zur Verfügung, wie zum Beispiel das Setzen von Kommentaren. Eine weitere wichtige Funktion ist die Extraktion von Textteilen, um diese dann in anderen Applikationen zu verwenden. So können zum Beispiel Daten aus einem Formular extrahiert werden, um sie dann in ein anderes Formular wieder einzufügen. Das Zusammenfügen und Trennen von einzelnen oder mehreren PDFs ist ebenso möglich, wie das Konvertieren einer PDF in ein Bildformat. Eine Besonderheit ist die mögliche Integration der Lucene Suchmaschine, welche ebenfalls ein Open Source Produkt ist. Mit Hilfe dieser ist das Indexing von PDF-Dokumenten möglich.

Für Apache PDFBox wurde eine eigene Website eingerichtet, die viele Informationen zur Implementierung und zu den verschiedenen Funktionalitäten beinhaltet. Hier hat der User die Möglichkeit, abhängig von seinen Interessen, sich für eine Art Newsletter einzuschreiben, über den er dann über Neuerungen informiert wird. Gleichzeitig

⁷ Vgl. The Apache Software Foundation (2013)

besteht aber auch die Möglichkeit sich an eine angegebene E-Mail-Adresse bei Fragen zuwenden.

Zusammenfassung Apache PDFBox:

- Guter Support über eigene Website
- Aktive Community
- Viele verschiedene Funktionen bereits integriert

4.2 icePDF⁸

icePDF ist wie alle anderen vorgestellten Produkte ebenfalls Open Source und Java-basiert. Hierbei ist allerdings zu erwähnen, dass die Open Source Version lediglich eine Grundversion ist und nicht alle Funktionen enthält. Im Gegensatz zu der kommerziellen Pro Version, die „voll ausgestattet ist“. Laut Herausgeber soll das Bearbeiten von PDF-Dokumenten mit diesem Produkt leicht und schnell möglich sein. Ebenso wie bei Apache PDFBox gibt es auch bei dieser Software bereits einige integrierte Funktionen, die verwendet werden können, wie zum Beispiel das Konvertieren von PDFs in Bildformate, die Extraktion von Textabschnitten oder Bilder, sowie die Suche und das Drucken von PDFs.

Eine besonders vielfältig einsetzbare Funktion ist das Setzen von Anmerkungen. Hier bietet das Produkt verschiedene Möglichkeiten. So können Anmerkungen hinzugefügt, verändert oder entfernt werden. Die Arten der Anmerkungen, die gesetzt werden können beinhalten das Markieren von Textpassagen. Text und Link Anmerkungen können gesetzt werden, ebenso wie Popup Anmerkungen auf die geantwortet werden kann.

icePDF kann als alleinstehender Java PDF Viewer verwendet werden oder auch ebenso einfach in eine andere Java Anwendung integriert werden. Auch für diese Software gibt es eine eigene Website, auf der eine Vielzahl von Dokumentationen bereitgestellt ist, wie zum Beispiel verschiedene Tutorials oder den Link zu einem Blog mit weiteren Informationen zum Produkt. Der Support ist allerdings nur für die Pro Version wirklich umfangreich.

⁸ Vgl. ICESOFTECHNOLOGIES INC. (O.J.)

Zusammenfassung icePDF:

- Demo Material verfügbar
- Guter Support über eigene Website
- Umfangreicher kommerzieller Support
- Viele verschiedene Funktionen bereits integriert
- Vielzahl von Annotation Funktionen

4.3 iText⁹

iText ist ein weiteres Produkt, um PDF-Dokumente zu bearbeiten. Wie bei icePDF muss hier aber in die Open Source Version und die Pro Version unterteilt werden. Die Open Source Version enthält auch hier nicht alle Funktionen. Der Fokus bei dieser Software wurde auf die Automatisierung der Bearbeitung gelegt. Dokumente können neu erstellt werden, zum Beispiel aus Daten von einem XML File oder aus Daten, die aus einer Datenbank bezogen werden. Existierende PDF-Dokumente können gelesen und bearbeitet werden. Der PDF Viewer ist allerdings nur in der kommerziellen Version vorhanden. Der Funktion des Lesens ist das Extrahieren von Textpassagen untergeordnet und für das Bearbeiten stehen Optionen wie stempeln, verschlüsseln, automatisches Ausfüllen von Formularen oder auch das Setzen von Unterschriften oder Wasserzeichen zur Verfügung. Auch dies sind Funktionen die nur in der kommerziellen Version verfügbar sind.

Mit dieser Software ist es aber auch möglich, verschiedene Schichten (Layer) über das Originaldokument zu legen. Das bedeutet, dass zum Beispiel ein Kommentar nicht in dem Ursprungsdokument gespeichert wird, sondern in einem Extra-Layer, das dann über das eigentliche Dokument gelegt wird. Das ermöglicht das Archivieren der Dokumente in ihrer ursprünglichen Form, ohne Kommentare oder Schwärzungen. Support für das Produkt ist ebenfalls über eine eigene Website gegeben. So hat der User zum Beispiel die Möglichkeit Fragen zu technischen Aspekten über eine Mailing Liste zustellen. Auch hier ist der Support für die kommerzielle Version deutlich umfangreicher.

⁹ Vgl. iText Software Corp. (o.J.)

Zusammenfassung iText:

- Java-Kenntnisse (auch in C# erhältlich)
- Support über eigene Website
- Viele verschiedene Funktionen bereits integriert
- Einbindung von Layern ist möglich

4.4 PDF Clown

PDF Clown ist eine Klassenbibliothek zur Manipulation von PDF-Dokumenten. Sie eignet sich daher vor allem für Softwareentwickler und nicht für gewöhnlichen PC Nutzer.¹⁰ Verfügbar ist sie für Java Version 6 und .Net Version 4.0.¹¹

Die Bibliothek kann aufgrund ihrer Fähigkeiten als PDF Generator, PDF Reader, PDF Editor und als PDF Rasterizer klassifiziert werden. Wobei letzteres bedeutet, dass die Bibliothek das Darstellen und das Drucken von Inhalten ermöglicht. Was bislang fehlt, möglicherweise aber in Zukunft ergänzt wird, ist die Klassifizierung als PDF Viewer.¹²

Unterstützt werden Anwender und Entwickler der Bibliothek durch eine Vielzahl an Kommunikationsmöglichkeiten. Diese werden mehrmals monatlich von der Community genutzt, was auf eine eher aktive Community schließen lässt. Es gibt zunächst zwei Foren, einen eigens für PDF Clown eingerichteten Blog, die Möglichkeit Bugs zu berichten und die Möglichkeit neue Funktionen vorzuschlagen. Darüber hinaus ist es sogar möglich, Fragen, Anregungen, etc. auf Twitter zu posten.

Zusammenfassung PDF Clown:

- Community ist mehrmals im Monat aktiv
- Unterstützt Java und .Net
- Kein PDF Viewer
- Umfangreiche Funktionalitäten
- Einbindung von Layern ist möglich

¹⁰ Vgl. Chizzolini, S. (2013a)

¹¹ Vgl. Chizzolini, S. (2013b)

¹² Vgl. Chizzolini, S. (2013a)

4.5 PDFjet

PDFjet ist eine für Java und für .Net verfügbare Klassenbibliothek zur dynamischen Erstellung von PDF-Dokumenten.

Der Hersteller bietet die Bibliothek in zwei verschiedenen Versionen an. Zum einen als kostenfreie Open Source Variante und zum anderen als kostenpflichtige Version.

Hinsichtlich des Funktionalitätsumfangs verfügt die Open Source Variante nur über grundlegende Funktionalitäten zur Erstellung von PDF-Dokumenten und bietet somit lediglich einen Bruchteil der Funktionalitäten der kostenpflichtigen Variante.¹³ Es werden bei der Open Source Variante beispielsweise nur zwei Schriftarten unterschieden.

Auf der Homepage von PDFjet existiert weder ein Blog, noch ein Diskussionsforum. Auch in anderen öffentlichen Blogs oder Foren wird selten über die Bibliothek geschrieben.

Zusammenfassung PDFjet:

- Verfügbar für Java und .Net
- Als Open Source und kostenpflichtige Variante verfügbar
- Nur grundlegende Funktionalitäten
- Keine Community

4.6 TCPDF

TCPDF ist eine PHP Klassenbibliothek zur Erzeugung von PDF-Dokumenten. Veröffentlicht ist die Bibliothek unter der LGPL Version 3.¹⁴

Die Klassenbibliothek bietet eine Vielzahl von Funktionalitäten, wie beispielsweise die Möglichkeit, Annotationen hinzuzufügen oder auch Lesezeichen zu setzen.¹⁵

Ähnlich zu der Klassenbibliothek PDF Clown wird auch für diese Bibliothek eine Vielzahl an Kommunikationsmöglichkeiten bereitgestellt. Es gibt ein Projektforum für

¹³ Vgl. Innovatics Inc. (2013) und Findbestopensource.com (2010)

¹⁴ Vgl. Asuni, N. (2013a)

¹⁵ Vgl. Asuni, N. (2013b)

Diskussionen und die Möglichkeit Bugs und Patches zu melden und Funktionalitätsanfragen zu stellen.¹⁶ Die Community ist eher aktiv, da sie grob geschätzt alle paar Tage aktiv ist.

Zusammenfassung TCPDF:

- Verfügbar für PHP, nicht für Java
- Umfangreiche Funktionalitäten
- Aktive Community

¹⁶ Vgl. Asuni, N. (2013c)

5 Bewertung und Auswahl des geeigneten Produktes

In dem folgenden Kapitel werden die zuvor vorgestellten Open Source Produkte hinsichtlich verschiedener Kriterien untersucht. Die Kriterien werden zunächst in Kapitel 5.1 definiert und gewichtet und in Kapitel 5.2. findet die erwähnte Untersuchung der einzelnen Produkte statt. Kapitel 5.3. gibt eine abschließende Bewertung und es wird eine Entscheidung für ein Produkt bzw. eine Kombination aus mehreren Produkten getroffen und begründet.

5.1 Vorstellung und Gewichtung des Kriterienkatalogs

Der nachfolgende Kriterienkatalog dient der Bewertung der in Kapitel 4 vorgestellten Bibliotheken. Er ergibt sich zum einen aus den in Kapitel 2.2 erörterten Anforderungen des Kunden und zum anderen aus Merkmalen von Open Source Produkten. Erste beziehen sich fast ausschließlich auf die gewünschte Funktionalität des Prototyps. Letztere müssen deshalb mitaufgenommen werden, da sie ausschlaggebend dafür sind, wie effizient und einfach die funktionalen Anforderungen umgesetzt werden können. Weitere Kriterien, wie beispielsweise Systemvoraussetzungen entfallen, da es sich bei dem Prototyp um ein Applet handeln soll.

Die jeweiligen Farben der Umrandung stellen eine Gewichtung hinsichtlich der Bedeutung der Kriterien dar. Rot bedeutet, sehr relevant, gelb steht für eine mittlere Wichtigkeit und grün dafür, dass das Kriterium von geringer Bedeutung ist. Die rot umrandeten Kundenanforderungen sind jene, die vom Kunden auf jeden Fall gefordert und somit unerlässlich für den Prototypen sind. Die weiteren wurden vom Kunden als optional eingestuft und sind daher von geringerer Relevanz.

Ebenso wichtig, wie die geforderten Funktionalitäten, sind die Qualität und der Umfang der Dokumentation der Bibliotheken, da diese zur Implementierung des Prototyps essentiell ist. Ein vorhandener Support, sowie eine aktive Community sind weitere hilfreiche Aspekte, jedoch nicht so unentbehrlich, wie eine gute Dokumentation. In Abbildung 1 wird der Kriterienkatalog noch einmal zusammengefasst dargestellt.

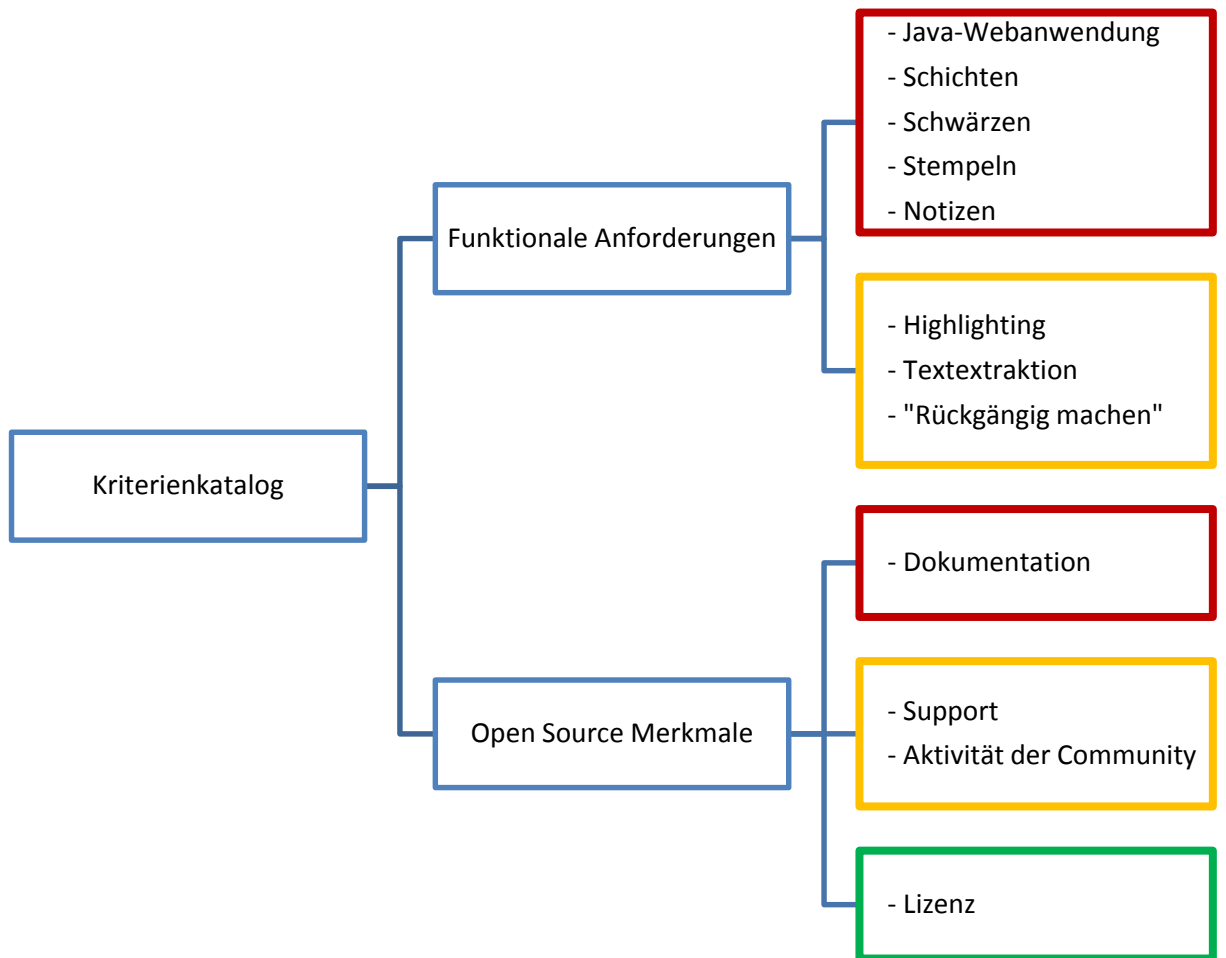


Abbildung 1: Kriterienkatalog

5.2 Bewertung der Libraries anhand des Kriterienkatalogs

Im Folgenden werden die vorgestellten Open Source Libraries anhand der zuvor definierten Kriterien bewertet.

Kunden- anforderungen	Apache PDFBox	icePDF	iText	PDF Clown	PDFjet	TcPDF
Programmiersprache	Java	Java, C#	Java	Java, .Net	Java, .Net	PHP
PDF-Ansicht	+	+++	X	X	X	X
Stempeln	X	X	++	+	X	+
Schwärzen	X	+	X	+	+	++
Kommentare	+	++	++	++	+	++
Extrahieren	++	+	X	+	X	X
Highlighting	X	+	X	+	X	++
Layer	X	X	+++	+	X	X
Rückgängig machen	X	X	X	X	X	X
Summe	3	8	7	7	2	(7)

Tabelle 1: Bewertung der Funktionalitäten

Open Source Merkmale	Apache PDFBox	icePDF	iText	PDF Clown	PDFjet	TcPDF
Lizenz	Apache License V2.0	MPL	Affere General Public License (AGPL).	Creative Commons Attribution-ShareAlike 2.5 License	Eigene Lizenz: PDFjet for Java and C# - Open Source Edition License	GNU Lesser General Public License V3
Dokumentation	Nur sehr grundlegende Beispiele, kaum weitergehend	Ausreichend vorhanden, mit einigen Beispielen	Dokumentation eingeschränkt (sehr gutes, aber kostenpflichtiges E-Book)	recht umfassend	Nur sehr grundlegende Beispiele, kaum weitergehend	Ausreichend vorhanden
Support	Mailing-Listen	kostenfreier Support nur begrenzt	Mailing-Listen, kostenfreier Support nur begrenzt	Foren, Blog, Feature und Bug Tracker	Nicht vorhanden	Forum, Bug, Patch und Feature Tracker,
Aktivität der Community	Aktives Forum	Beschränkt, kommerziell gelenkt	Beschränkt, kommerziell gelenkt	Sehr aktiv	Kaum aktiv	Sehr aktiv

Tabelle 2: Zusammenfassung der Open Source Merkmale

5.3 Auswahl einer geeigneten Java Library

Die beiden Tabellen in Kapitel 5.2 stellen bereits eine umfassende Zusammenfassung der Untersuchung der Open Source Produkte hinsichtlich der definierten Kriterien dar. In diesem Abschnitt wird nun anhand der Gewichtung der Kriterien eine Entscheidung für ein Produkt oder für die Kombination mehrerer Produkte getroffen. Hierzu werden die Ergebnisse zunächst einmal miteinander verglichen.

Apache PDFBox ist von den ausgewählten Produkten am besten geeignet, um die Funktion des Extrahierens umzusetzen und ist neben icePDF das einzige, das eine PDF Ansicht anbietet. Im Gegenzug dazu ist es aber nicht möglich, Layer miteinzubauen, welches als ein sehr wichtiges Kriterium gewertet wird. Wie bereits erwähnt setzt icePDF eine PDF Ansicht um, allerdings noch um einiges besser als es mit Apache PDFBox möglich ist. Ein sehr großer Pluspunkt für dieses Produkt ist die gute GUI, die ebenfalls mitgeliefert wird. Im Vergleich zu den anderen Produkten ist icePDF das Einzige, das dies ermöglicht. Ein weiterer wichtiger Aspekt ist die gute Dokumentation.

iText ist neben PDF Clown die einzige Software, die es ermöglicht Layer miteinzubinden, was eines der Hauptkriterien darstellt, jedoch ist die Funktion bei iText wesentlich ausgereifter. Das Einfügen von Kommentaren und das Setzen von Stempeln lässt sich mit diesem Produkt gut umsetzen. Hierbei besteht allerdings, wie auch bei icePDF das Problem, dass die Aktivität der Community für die Open Source Version sehr beschränkt ist und vom Hersteller gelenkt wird. PDF Clown setzt alle wichtigen Kriterien gut um und ermöglicht auch den Einsatz von Layern. Auch die Community für dieses Produkt ist sehr aktiv.

PDFjet stellt in der freien Version lediglich das Setzen von Kommentaren und das Schwärzen von Textpassagen zur Verfügung. Die kostenpflichtige Version hingegen ermöglicht deutlich mehr Funktionen. Hinzu kommt, dass die Community für PDFjet kaum aktiv ist. TcPDF ist außerhalb der Wertung, da es das Kriterium der Programmiersprache (Java) nicht erfüllt. Ansonsten sind auch hiermit viele der geforderten Funktionen möglich und es könnte eine gute Alternative zu einem Java-basierten Produkt bieten.

Anhand der Kriterien wurde entschieden, für die Umsetzung eine Kombination aus icePDF und iText zu wählen. Zwar enthält die Open Source Version von icePDF we-

niger Funktionen als die kommerzielle Version und ebenso verhält es sich mit dem Support. Dennoch sind in der Open Source Version alle für diese Projekt notwendigen Funktionen, bis auf das Stempeln und das Verwenden von Layern, enthalten. Diese können aber ergänzt werden. Dazu kommt die sehr gute GUI, die bereits integriert ist. Die Layer-Funktion ist sowohl mit iText, als auch mit PDF Clown umsetzbar. Allerdings ist die Dokumentation dafür bei iText deutlich besser. PDF Clown kann auch als Alternative zu icePDF verwendet werden, hier fehlt aber die Viewerfunktion.

6 Dokumentation der Implementierung des Prototypes

Nach der Analyse und Auswahl von geeigneten Java-Libraries zur Erstellung einer Webanwendung zur Bearbeitung von PDF-Dateien soll nun auf die Implementierung des Prototypes eingegangen werden. Zunächst soll der geplante Aufbau der Webanwendung aufgezeigt werden. Anschließend werden dazu die Systemvoraussetzungen erläutert und die Entwicklungsumgebung beschrieben. Die bereits vorgestellten Anforderungen des Kunden an die Webanwendung werden dann dazu genutzt, benötigte Anpassungen der genutzten Java-Klassen aus den Bibliotheken aufzuzeigen. Abschließend sollen die Lösungen vorgestellt werden, wie die Java-Klassen entsprechend den Vorgaben bearbeitet wurden.

6.1 Aufbau der Webanwendung

Hauptbestandteil der Webanwendung werden die Libraries icePDF und iText sein. Zusätzlich zu den vorhandenen Klassen und Methoden sollen weitere Klassen entwickelt, die einerseits für den Aufruf des Bearbeitungsfensters und andererseits für Durchführung der Bearbeitung selbst dienen sollen. Um die Webanwendung über den Browser zu öffnen, muss der Nutzer auf die Java Server Page `index.jsp`, die in der folgenden Abbildung 2 gezeigt wird, navigieren und dort das Bearbeitungsfenster über das Anklicken des Buttons starten.



Abbildung 2: Ansicht der `index.jsp`

Das sich öffnende Fenster zeigt zunächst nur die Werkzeuge an, welche zur Bearbeitung genutzt werden können. Über den Button „Öffnen“ kann eine existierende PDF ausgewählt werden, sodass diese aus der verbundenen relationalen Datenbank geladen wird. Hierbei wird nicht nur die Original-Datei, sondern alle bereits stattgefundenen Änderungen hinzugefügt, indem während des Vorgangs eine Abfrage an die Datenbank nach eventuellen Änderungen durchgeführt wird. Diese können wahlweise über den Menüpunkt „Layer“ ausgeblendet werden, um die Übersichtlichkeit der Datei zu erhöhen und die Originaldatei betrachten zu können.

Nimmt der Nutzer weitere Änderungen in der Webanwendung vor, werden diese Änderungen auf die entsprechende Schicht gelegt und die Ansicht wird aktualisiert. Im Hintergrund werden die Änderungen in der Datenbank vermerkt, sodass weitere Benutzer diese Änderungen ebenfalls anschauen und nutzen können.

Zur Illustration der Architektur der Webanwendung soll diese Kapitel in der folgenden Abbildung 3 zusammengefasst werden.

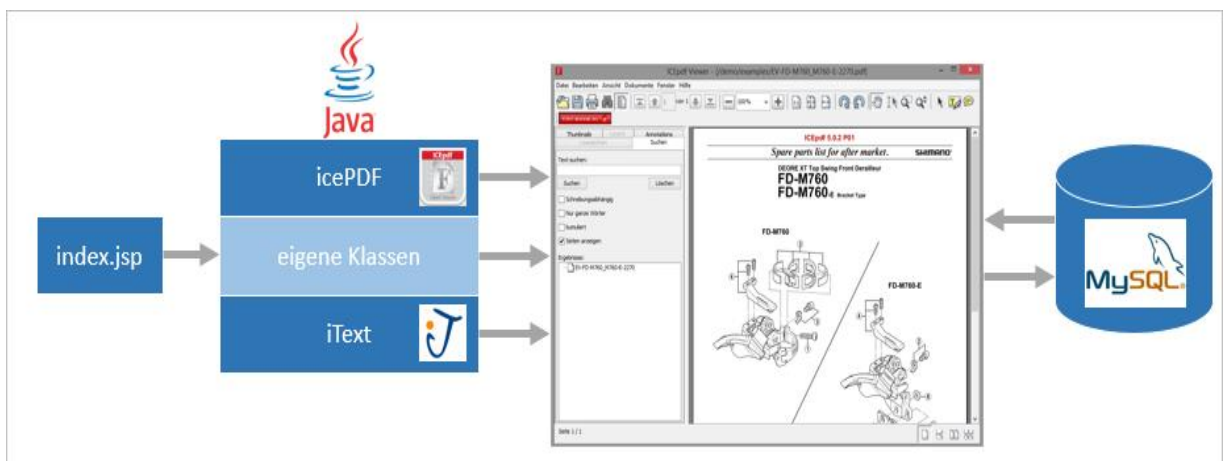


Abbildung 3: Aufbau der Webanwendung

6.2 Vorstellung der Systemvoraussetzungen

Für den Einsatz des Prototypens im Umfeld des Kunden werden ein Webserver, sowie ein Java Development Toolkit benötigt. Zusätzlich wird eine relationale Datenbank zum Speichern der editierten PDF-Dateien genutzt, bei welcher es sich aufgrund der fehlenden Lizenz im Rahmen der Entwicklung jedoch nicht um die im Unternehmen im Einsatz befindliche DB2-Datenbank handelt. Stattdessen wird in Ab-

sprache mit dem Kunden das Open Source Datenbankmanagementprodukt MySQL eingesetzt. Für den Einsatz der Webanwendung in der IT-Umgebung des Kunden bedarf es folglich der Anpassung an die genutzte Datenbankverwaltungssoftware.

Neben der beim Kunden benötigten Software stellt die folgende Tabelle 3 außerdem die in der Entwicklung genutzten Produkte dar, um die Umsetzung besser nachvollziehen zu können. Zusammen mit NetBeans als Entwicklungsumgebung wird Apache Tomcat als Webserver für die zu entwickelnde Anwendung eingesetzt. Die Java-Bibliotheken icePDF und iText werden in der aktuellsten stabilen Version eingesetzt und in die Webanwendung integriert. Für den Zugriff auf die relationale Datenbank wird die Software MySQL Connector/J benutzt. Entsprechend der Nutzung im Unternehmen ist die Webanwendung für den Internet Explorer als Browser ausgelegt.

Software	Version	Release Datum
NetBeans IDE	7.2.1	06.11.2012
Apache Tomcat Webserver	7.0.27.0	05.04.2012
JDK	1.7.0.21	16.04.2013
icePDF	5.0.1	08.05.2013
iText	5.4.2	31.05.2013
MySQL	5.6.11	18.04.2013
MySQL Connector/J	5.1.25	01.05.2013
Internet Explorer	9.0.17	11.06.2013

Tabelle 3: Übersicht der eingesetzten Software

6.3 Benötigte Anpassungen der Libraries

Die von icePDF mitgelieferte GUI (Graphical User Interface) dient bei der Entwicklung der Webanwendung als erster Ansatzpunkt, da in dieser Ansicht sowohl bereits das PDF-Dokument an sich, als auch ein Menü mit den unterschiedlichen Bearbeitungswerkzeugen zur Verfügung gestellt werden. Um die intuitive Bearbeitung der Dokumente zu erleichtern soll neben der horizontalen Symbolleiste oben im Fenster die detaillierte Ansicht vom Start der Anwendung geöffnet zur Verfügung stehen. Die ursprüngliche Benutzungsoberfläche wird in der folgenden Abbildung 4 dargestellt.

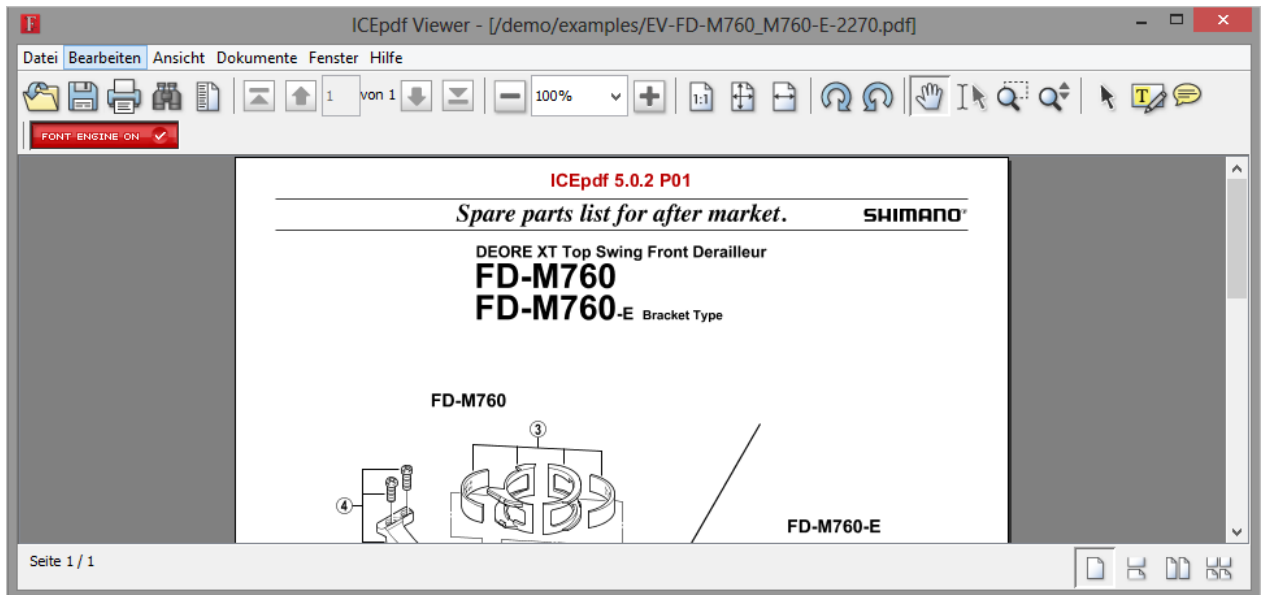


Abbildung 4: Benutzungsoberfläche von icePDF¹⁷

Zudem sollen zur Vereinfachung alle nicht benötigten und damit auch nicht vollständig in die Webanwendung integrierten Funktionen aus der Werkzeugauswahl entfernt werden. Für die vom Kunden gewünschten Funktionen ist vor allem zu beachten, dass die bearbeiteten Elemente jeweils auf unterschiedlichen Schichten eingefügt werden, sodass die einzelnen Elemente je nach Bedarf ein- und ausgeblendet werden können. Dies stellt den größten Aufwand der Anpassungen dar, da hier eine Verknüpfung der bereits vorhandenen Grundfunktionen von icePDF, wie z.B. das Einfügen von Notizen oder von Rechtecken (zum Schwärzen) mit der Schichtenfunktionalität von iText erarbeitet werden muss. Die GUI unterstützt bereits die optionale Ansicht von Schichten, wenn diese in der zu öffnenden PDF-Datei vorhanden sind.

Zum Einfügen eines Stempels besteht in icePDF keine nutzbare Funktion, sodass hier vollständig auf die Klassen und Methoden von iText zurückgegriffen werden muss. Das Einfügen von Bildern in vorhandene PDF-Dateien soll hierfür so abgeändert werden, dass ein Stempelmotiv auf der PDF eingefügt wird, welches einen variablen Teil, das aktuelle Datum, enthält. Vergleichbar mit den vorherigen Funktionen sollen auch die Stempel auf einer jeweils neuen Schicht eingefügt werden. Für den Prototyp wird zunächst nur beispielhaft die Nutzung eines Stempels implementiert. Die Erweiterung der Implementierung um weitere, optionale Stempel soll anhand der Dokumentation des Prototypes ermöglicht werden.

¹⁷ <http://www.icesoft.org/java/demos/icepdf-demos.jsf>

Abschließend bleibt festzuhalten, dass in der Open Source Variante von icePDF die Funktion zum Speichern der bearbeiteten PDF-Datei deaktiviert ist, sodass hier eine eigene Logik entwickelt werden muss. Da die Dokumente des Kunden in einer Datenbank gespeichert werden, sollen auch die Änderungen darin eingetragen werden. Für den Prototypen sollen somit alle getätigten Änderungen in einer relationalen Datenbank hinterlegt werden. Wird das bearbeitete Dokument erneut aufgerufen sollen anhand der Informationen der Datenbank die Änderungen erneut durchgeführt und sozusagen „abgespielt“ werden, sodass dem Nutzer letztendlich die PDF wieder wie vor dem Speichern zur Verfügung steht. Der Lösungsansatz eine Datenbank für das Editieren der PDF-Dokumente zu nutzen wird von keiner der beiden ausgewählten Java-Libraries unterstützt und muss vollständig selbst entwickelt werden.

6.4 Implementierung des Prototyps

Nach der Planung des Prototyps soll sich der Fokus nun auf die konkrete Implementierung richten und die wichtigsten Anpassungen dokumentieren. Die Unterteilung dieses Kapitels erfolgt nach den Arbeitsschritten bei der PDF-Editierung beginnend mit dem Zugriff auf die Datenbank, über die Nutzung der verschiedenen Funktion bis zum Speichern des fertigen Dokumentes.

6.4.1 Aufbau der Datenbank und Zugriff darauf

Da der Fokus der Arbeit auf der Entwicklung der Webanwendung mit den Bearbeitungsfunktionen und weniger auf der Datenbank und dem Abspeichern der Dokumente liegen soll, wird ein vereinfachtes Datenbankschema angelegt, um die Grundfunktionalitäten demonstrieren zu können. Für jede Bearbeitungsfunktion wird eine eigene Tabelle erstellt, deren Spalten den Eigenschaften der Änderung entsprechen. Neben der eindeutigen ID wird jeder Änderung außerdem eine Spalte mit der ID des passenden PDF-Dokumentes als Fremdschlüssel zugewiesen. In einer zusätzlichen Tabelle werden die bearbeiteten Dokumente mit der als Referenz dienenden ID und dem Dateipfad gespeichert, sodass die Anwendung über den Dateipfad das Dokument öffnen kann. Das Datenbankschema mit allen enthaltenen Tabellen, sowie deren Spalten und Datentypen wird in Abbildung 5 dargestellt.

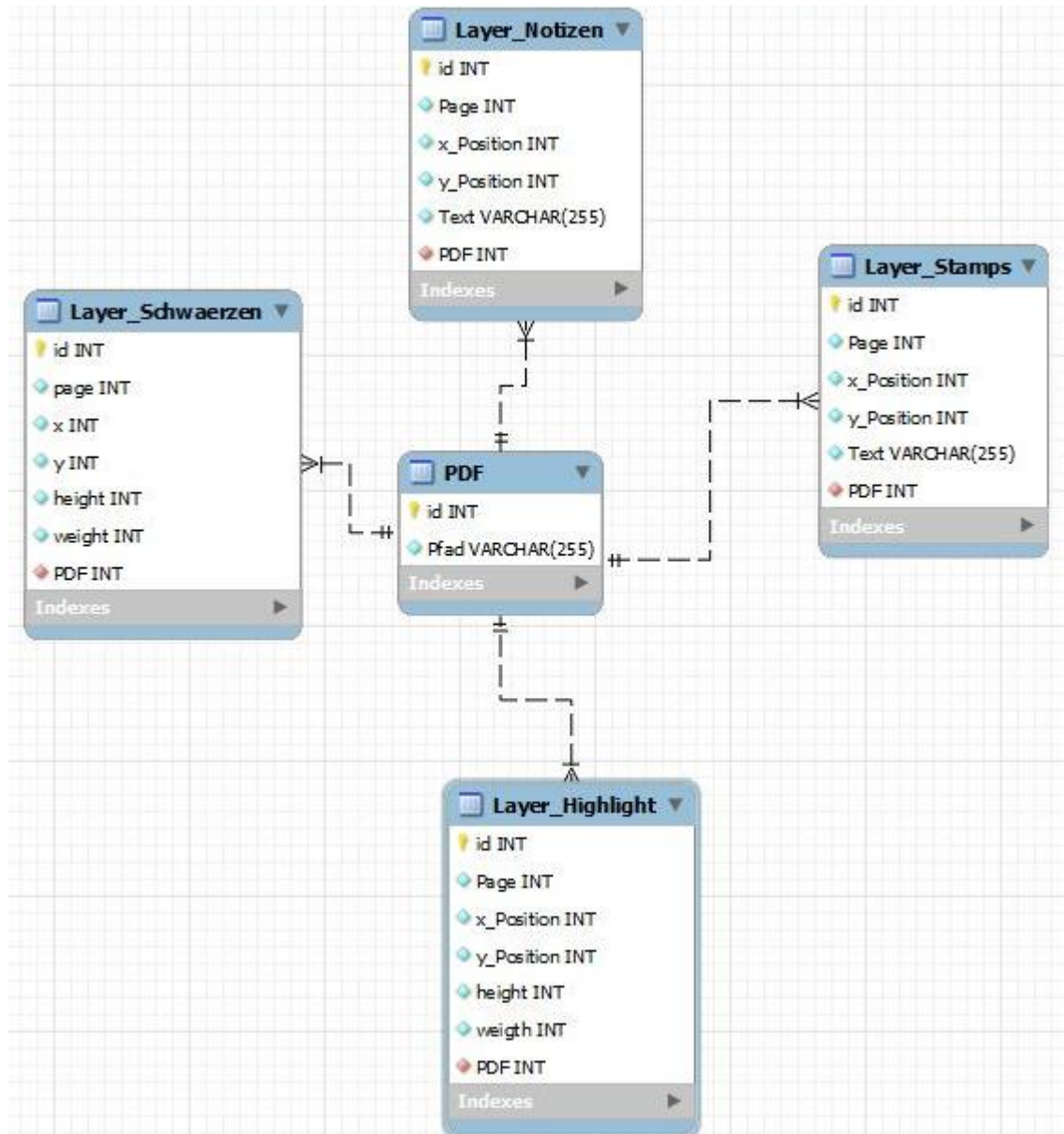


Abbildung 5: Verwendetes Datenbankschema

Die importierte Bibliothek des Datenbanktreibers MySQL Connector/J beinhaltet die Klassen zum Zugriff auf die Datenbank mit der Java-Webanwendung. Damit die Anwendung übersichtlich bleibt, werden alle Methoden im Zusammenhang mit dem Datenbankzugriff, d.h. vom Herstellen der Verbindung über Abfragen bis hin zum Speichern von Änderungen in der Datenbank, in dieser Klasse „*JDBC.java*“ zusammengefasst. Diese Methoden können dann innerhalb der Bearbeitungsfunktionen mit den jeweiligen auszuführenden SQL-Statements aufgerufen werden.

6.4.2 Erstellung der Schichten

Für jede Art der Änderungen, beispielweise für das Hinzufügen von Notizen, wird eine eigene Schicht in die PDF-Datei hinzugefügt. Sobald die erste Bearbeitung stattfindet werden dazu alle benötigten Schichten innerhalb der Klasse „*Layering*“ erstellt. Neben der Erstellung der Schichten beinhaltet diese Klasse auch alle Informationen über bereits hinzuegefügte Anmerkungen, d.h. für jede Bemerkung existiert ein Array, in welchem die Objekte mit ihren Attributen gespeichert werden. Das Array „*RectangleArray []*“ beispielsweise beinhaltet sämtliche mit Hilfe der in Kapitel 6.4.3 erklärten Funktion erstellten Rechtecke zum Schwärzen. Zusätzlich speichert das Array „*RectanglePositionArray []*“ die x- und y-Koordinaten des Rechtecks auf der PDF-Datei, sowie die Seitenzahl ab.

Neben den getter- und setter-Methoden für den Lese- und Schreibzugriff auf die Arrays ist die Methode „*setWriter*“ von zentraler Bedeutung, da in dieser Methode die eigentliche Bearbeitung der PDF-Datei stattfindet. Es werden jeweils die entsprechenden Schichten (Schwärzen, Notizen, Stempel und Highlight) zum Dokument hinzugefügt und die Objekte aus den Arrays, die die Informationen über die Bearbeitungen enthalten, auf die PDF „geschrieben“. Der folgende Code-Ausschnitt soll der Verdeutlichung dienen.

```
//Methode zum Erstellen und Beschreiben der Layers
public static void setWriter()
{
    try{
        fos = new FileOutputStream(filename);
        reader = new PdfReader(filePath);
        stamper = new PdfStamper(reader, fos);

        PdfContentByte cb;

        //erstellt neue Layer für Schwärzungen
        wmLayerBlack=new PdfLayer("Schwaerzungen",stamper.getWriter());

        //zeichnet jedes Rechteck aus dem Array auf die Layer
        for(int i=0; i<RectangleArray.length;i++)
        {
            //Schleife abbrechen, falls kein Objekt mehr vorhanden ist
            if(RectangleArray[i]==null)
                break;

            //Rechteck wird über dem Inhalt erstellt
            cb = stamper.getOverContent(RectanglePositionArray[i]+1);

            cb.beginLayer(wmLayerBlack);
```

```

//Rechteck aus Array hinzufügen
Rectangle draw=RectangleArray[i];
cb.rectangle(draw.x,draw.y,draw.width,draw.height);
cb.fill();
cb.endLayer();
}

//erstellt neue Layer für Notizen
wmLayerText=new PdfLayer("Notizen",stamper.getWriter());
//for-Schleife siehe oben

//erstellt neue Layer für Stempel
wmLayerStempel=new PdfLayer("Stempel", stamper.getWriter());
//for-Schleife siehe oben

stamper.close();
reader.close();
fos.close();
}

catch(Exception ea)
{
    //...
}
}

```

6.4.3 Funktion „Schwärzen“

Für das Schwärzen eines Bereichs in einem PDF-Dokument wird die von icePDF bereitgestellte Funktion zum Einfügen von Rechtecken genutzt. Diese ist über den in der nebenstehenden Abbildung 6 markierten Button zu erreichen. Die Standardwerte wurden hierfür zunächst auf ein ausgefülltes, schwarzes Rechteck geändert. In der Klasse „*SquareAnnotationHandler*“¹⁸ finden sich die Methoden zur Anpassung dieser Funktion. Relevant für die Umsetzung der Schichten-Struktur der PDF-Datei ist hier die Methode zum Einfügen des Rechteckes beim Loslassen der Maustaste, nachdem der Nutzer per Drag-and-Drop das Rechteck in die gewünschte Größe gezogen hat. In der Methode „*public void mouseReleased(MouseEvent e)*“ werden die Attribute des entstehenden Rechteckes (ID, x- und y-Position, Höhe, Breite und Seitenzahl) an die Klasse „*Layering*“ übergeben, in welcher dann die Eigenschaften des Rechteckes im zugehörigen Array gespeichert werden.

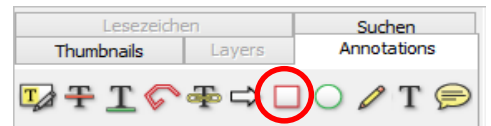


Abbildung 6: Button zum Schwärzen

¹⁸ Klasse enthalten in: org.icepdf.ri.common.tools

Abschließend werden in der Klasse die Methode „*Layering.setWriter()*“ und weitere Befehle zum Aktualisieren der PDF-Ansicht eingefügt, um die neu erstellte Schicht und die dazugehörigen Rechtecke zu erstellen und einblenden zu können.

6.4.4 Funktion „Notizen einfügen“

Der in Abbildung 7 hervorgehobene Button dient zum Einfügen eines neuen Textfeldes um Notizen zu einer bestehenden PDF-Datei hinzuzufügen zu können. Der Aufbau der Funktion ist vergleichbar mit der zuvor erklärten: In der Klasse „*FreeTextAnnotationHandler*“¹⁹ wird in der entsprechenden Methode „*public void mouseReleased(MouseEvent e)*“ der Aufruf einer Dialogbox eingefügt, in welche der Benutzer seine gewünschte Notiz einfügen kann. Nach dem Beenden der Eingabe werden dann die Eigenschaften der Notiz, also ID, x- und y-Position, der Text der Notiz und die Seitenzahl, an die Klasse „*Layering*“ übergeben und die Ansicht neu geladen.

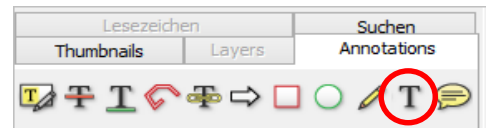


Abbildung 7: Button zum Einfügen von Notizen

6.4.5 Funktion „Stempeln“

Während die bisherigen Funktionen mit Ausnahme des Hinzufügens der neuen Schicht auf Klassen aus icePDF basieren, ist die Stempelfunktion auf Grundlage von iText-Klassen entwickelt worden. Dennoch ist die Funktionsweise auf derselben Logik aufbauend wie die bisher beschriebenen Funktionen. Statt dem Einfügen einer Form oder eines Textes wird in diesem Fall ein Bild geladen und dieses auf der Schicht eingefügt. Zusätzlich wird in das Bild, welches den Stempel darstellt ein Text eingefügt, der beispielsweise für das Hinzufügen eines Datums geeignet ist und somit einen Eingangsstempel darstellen könnte. Der Button hierfür ist in der nebenstehenden Abbildung 8 abgebildet. Unterschiedliche Stempel können entsprechend durch verschiedene Bilder umgesetzt werden. Der Prototyp enthält zunächst nur einen Stempel, denkbar ist allerdings auch, dass vor dem Einfügen des Stempels eine Benutzerabfrage stattfindet, in welcher der Nutzer das Bild für den Stempel aus-

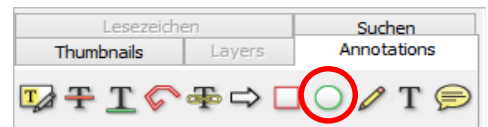


Abbildung 8: Button zum Einfügen eines Stempels

¹⁹ Klasse enthalten in: org.icepdf.ri.common.tools

suchen kann. Es existiert bereits eine Abfrage, in welcher der Nutzer eine Eingabe zum Stempel hinzufügen kann. Hier könnte alternativ auch das aktuelle Datum automatisch hinzugefügt werden. Die Änderungen des Quellcodes befinden sich in der Klasse „*CircleAnnotationHandler*“ in der entsprechenden Methode. Der Pfad für das Stempel-Bild ist in der Klasse „*Layering.java*“ in der Methode „*setWriter()*“ zu finden und kann beliebig geändert oder durch eine Benutzerabfrage ersetzt werden.

6.4.6 Weitere Funktionen

Für das Highlighting wird die Funktion zum Schwärzen von Text genutzt und so abgeändert, dass statt einem schwarzen Rechteck ein transparentes, blaues Rechteck eingefügt wird.

Die Auswahl findet über den Button, der in Abbildung 9 gezeigt wird, statt. Alle Änderungen wurden in der Klasse „*HighlightAnnotationHandler*“ eingefügt.

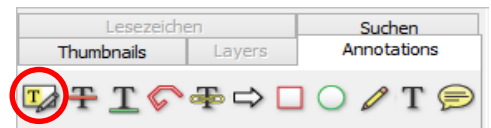


Abbildung 9: Button zum Highlighten eines Textes

Um eine sinnvolle Bearbeitung von PDF-Dokumenten zu ermöglichen werden neben den

Funktionen zum Hinzufügen von Inhalt auch eine Methode zum entfernen bzw. rückgängig

machen dieser benötigt. Im Prototyp wurde dies mit Hilfe eines mehrdimensionalen Arrays realisiert, in welchem die hinzugefügten Objekte bei ihrer Erstellung gespeichert werden. Durch Aktivierung des in Abbildung 10 dargestellten Buttons und erneuten Klicken auf das Dokument wird die Methode „*public void mouseReleased(MouseEvent e)*“ der Klasse „*InkAnnotationHandler*“ ausgeführt. Diese beinhaltet eine Methodenaufruf der „*undo()*“ Methode der Klasse *Layering*. In dieser wird aus dem erwähnten Array das letzte erzeugte Objekt gelöscht und die Ansicht neu geladen. Problematisch bei der Umsetzung erwies sich die Funktion zur Extraktion von Textteilen aus der PDF-Datei. Diese wird nicht anhand von Java-Klassen umgesetzt, sondern sollte laut Dokumentation von icePDF wie üblich mit Hilfe der normalen Kopieren-und-Einfügen Option möglich sein. Es stellte sich jedoch heraus, dass dies nicht fehlerfrei möglich ist. Der Nutzer kann zwar Text auswählen und kopieren, beim Einfügen wird jedoch der kopierte Text gleich mehrmals untereinander angezeigt. Im Rahmen des Prototypens konnte keine Lösung zur Behebung des Problems gefunden werden.

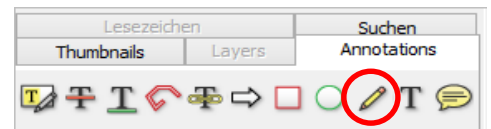


Abbildung 10: Button zum "Rückgängig machen"

Abschließend wurden alle nicht belegten Buttons aus der GUI ausgeblendet, um die Ansicht für den Nutzer zu verbessern.

6.4.7 Speichern der Änderungen in der Datenbank

In der Klasse „*Layering*“ befindet sich eine weitere Methode, welche zum Speichern des Dokumentes in der Datenbank dient. Der Nutzer kann diese Funktion wie üblich über das Disketten-Symbol aufrufen. Die Methode „*public static void save()*“ soll diesen Zweck erfüllen, in dem es zunächst die ID des geöffneten Dokumentes aus der Datenbank abfragt. Mit Hilfe dieser ID kann der folgende Quellcode je nach Art der Bearbeitung die entsprechende Tabelle ansprechen und dort die Änderungen abspeichern. Das folgende Beispiel bezieht sich auf die Funktion „Schwärzen“ und das Einfügen der entsprechenden Rechtecke. Die Eigenschaften des Rechtecks werden dann in die dafür vorgesehene Tabelle eingetragen, sodass das Rechteck eindeutig wiederhergestellt werden kann. In der Spalte PDF wird als letztes die ID des bearbeiteten PDF-Dokuments eingetragen, um die Beziehung zwischen dem Dokument und der Bearbeitung herzustellen. Der Befehl „*JDBC.InsertToDB()*“ verweist auf die Klasse „*JDBC*“, die die Verwaltung und Kommunikation mit der Datenbank behandelt und letztendlich den Insert-Befehl ausführt.

```
//For-Schleife, ausgeführt für jedes Rechteck auf dieser Schicht
for(int i = 0; i < RectangleArray.length; i++)
{
    //Ausführung wird beendet, wenn kein Objekt mehr enthalten ist
    if(RectangleArray[i] == null)
        break;

    Rectangle draw = RectangleArray[i];

    sql = "INSERT INTO mydb.layer_schwaerzen (page, x, y, height,
weight, PDF) values (";

    sql = sql + ((int)RectanglePositionArray[i]+1) + "," +
(int)draw.x + "," + (int)draw.y + "," + (int)draw.height + "," +
(int)draw.width + "," + id;

    sql = sql + ")";

    JDBC.InsertToDB(sql);
}
```

7 Fazit

Ziel der Arbeit ist es eine Open Source basierte Alternative zu der bestehenden kommerziellen Software zur Bearbeitung von PDF-Dokumenten des Kunden ausfindig zu machen und zu implementieren.

Hierfür wurden zunächst die genauen Anforderungen des Kunden geklärt und darauf basierend eine Onlinerecherche zur Auswahl geeigneter Software-Bibliotheken durchgeführt. Diese Recherche führte zu einer engeren Auswahl von sechs Bibliotheken, die anschließend anhand eines Kriterienkatalogs bewertet wurden. Auf Basis dieser Bewertung war es möglich die geeignetsten Bibliotheken zur Implementierung eines Prototyps auszuwählen.

Die Wahl fiel dabei auf eine Kombination der Bibliotheken icePDF und iText. Die erste Bibliothek wurde hauptsächlich aufgrund der mitgelieferten GUI und den umfangreichen „Annotation“-Funktionengewählt und die zweite aufgrund dessen, da sie die einzige ist, die eine den Anforderungen entsprechende Einbindung von Layern ermöglicht.

Anschließend wurde mit der Umsetzung des Prototyps begonnen. Dabei lag eine besondere Herausforderung darin, die zwei verschiedenen Produkte zu kombinieren und eine einheitliche Anwendung aufzubauen. Nachdem eine Programmierlogik zur Kombination einer Funktion aus icePDF und der Schichten-Struktur aus iText gefunden wurde, konnte für alle darauffolgenden Funktionen verwendet werden, sodass alle in der Arbeit beschriebenen Funktionen (Schwärzen, Notizen, Stempeln und Highlight) darauf beruhen. Das Extrahieren von Text war leider auch nicht umsetzbar. Zusätzlich konnte jedoch das „Rückgängig machen“ einer Änderung ermöglicht werden.

Ein weiterer Kritikpunkt besteht darin, dass die Performance noch verbessert werden muss, da die Ladezeiten beispielsweise bei der Funktion zum Einfügen von Notizen recht lang sind.

Abschließend besteht das Ergebnis der Arbeit jedoch in einem funktionsfähigen Prototyp, der die vom Kunden geforderten Funktionen Stempeln, Schwärzen und das Einfügen von Kommentaren erfolgreich erfüllt. Darüber hinaus ist es möglich alle genannten Funktionen mithilfe der geforderten Schichten umzusetzen.

Für eine Nutzung des Prototyps durch den Kunden müssten allerdings noch entsprechende Anpassungen stattfinden. Dazu gehört die Anbindung an die Datenbank des Kunden, sowie die Integration des Prototyps in das bestehende Dokumentenmanagementsystem.

Quellenverzeichnisse

Literaturverzeichnis

Keller-Stoltenhoff, E. (2008) Nutzungsrechte an Freier Software, München: IT-Recht Kanzlei

Verzeichnis der Internet- und Intranet-Quellen

- | | |
|-------------------------------|---|
| Adobe Systems (o.J.) | Adobe PDF, http://www.adobe.com/de/products/acrobat/adobepdf.html , Abruf: 30.06.2013 |
| Asuni, N. (2013a) | TCPDF Software Licence, http://www.tcpdf.org/license.php , Abruf: 20.06.2013 |
| Asuni, N. (2013b) | TCPDF is a FLOSS PHP class for generating PDF documents, http://www.tcpdf.org/ , Abruf: 20.06.2013 |
| Asuni, N. (2013c) | TCPDF – PHP Class for PDF – Support, http://sourceforge.Net/projects/tcpdf/support , Abruf: 20.06.2013 |
| Chizzolini, S. (2013a) | PDF Clown Features, http://www.stefanochizzolini.it/en/projects/clown/features.html , Abruf: 18.06.2013 |
| Chizzolini, S. (2013b) | PDF Clown, http://www.stefanochizzolini.it/en/projects/clown/ , Abruf: 18.06.2013 |
| Findbestopensource.com (2010) | PDFjet for Java – FAQ, http://pdfjet.com/java/faq.html , Abruf: 20.06.2013 |

- ICESOFT Technologies Inc. (o.J.) ICEpdf Overview, <http://www.icesoft.org/java/projects/ICEpdf/overview.jsf>, Abruf: 27.06.2013
- ifrOSS (o.J.a) Lizenz Center, <http://www.ifross.org/lizenz-center#term-219>, Abruf: 29.05.2013
- ifrOSS (o.J.b) Welches sind die wichtigsten Open Source Lizenzen und welchem Lizenztyp gehören sie an?, <http://www.ifross.org/welches-sind-wichtigsten-open-source-lizenzen-und-welchem-lizenztyp-gehoren-sie>, Abruf: 29.05.2013
- Innovatics Inc. (2013) PDFJet – PDF library for Java and .Net, <http://www.findbestopensource.com/product/pdfjet>, Abruf: 20.06.2013
- iText Software Corp. (o.J.) What is iText?, <http://itextpdf.com/itext.php>, Abruf:27.06.2013
- Lackes, R./ Siepermann, M. (o.J.) Open Source, <http://wirtschaftslexikon.gabler.de/Definition/open-source.html>, Abruf: 10.05.2013
- Open Source Initiative (o.J.) The Open Source Definition, <http://opensource.org/osd>, Abruf: 10.05.2013
- The Apache Software Foundation (2013) Apache PDFBox - A Java PDF Library, <http://pdfbox.apache.org/>, Abruf: 29.05.2013

Evaluation of Java-based TIFF processing approaches – realization of an
exemplary prototype for a German mid-size insurance company

Seminar paper

submitted on July 2, 2013

School of: Business

Program: International Business Information Management

Course: WWI2010I

by

Joe André Boden, Philip Fitschen and Max Vökler

Table of contents

Terminology list	IV
List of figures	V
List of tables	V
1 Introduction, problem description and approach	1
2 TIFF – specifications and challenges	2
2.1 Characteristics of TIFF	2
2.2 Challenges	4
3 Approaches to TIFF processing	7
3.1 Native Java (applet)	7
3.2 HTML5 and JavaScript technologies	7
3.3 Adobe Flash, Microsoft Silverlight.....	7
3.4 Frameworks	7
3.4.1 <i>ImageMagick®</i>	8
3.4.2 <i>GraphicsMagick</i>	8
3.4.3 <i>Netpbm</i>	8
3.4.4 <i>VIPS</i>	9
3.4.5 <i>Apache Commons Imaging</i>	9
4 Exemplary implementation	10
4.1 Requirements Engineering	10
4.1.1 <i>Purpose, scope and overview</i>	10
4.1.2 <i>Product perspective</i>	10
4.1.3 <i>Product functions (business requirements)</i>	10
4.1.4 <i>User characteristics</i>	11
4.1.5 <i>Constraints</i>	11
4.1.6 <i>Apportioning of requirements</i>	12
4.1.7 <i>Specification – External interfaces, product and performance requirements..</i>	12
4.1.8 <i>Specification – Component requirements</i>	15
4.1.9 <i>Specification – Logical database requirements</i>	15
4.2 Application of decision criteria for technology choice	15
4.3 Exemplary TIFF processing tool	16
4.3.1 <i>Software design</i>	17
4.3.2 <i>Software implementation</i>	17

5 Conclusion	19
List of references	21

Terminology list

API	A pplication P rogramming I nterface
IT	I nformation T echnology
JNI	J ava N ative I nterface
JSP	J ava S erver P ages
PDF	P ortable D ocument F ormat
TIFF	T agged I mage F ile F ormat
IFH	I mage F ile H ead
IFD	I mage F ile D irectory

List of figures

Fig. 1: TIFF illustrated	4
Fig. 2: Database schema	15

List of tables

Table 1: TIFF file header	2
Table 2: TIFF IFD parts	3
Table 3: TIFF IFD Entry parts.....	3
Table 4: Decision criteria for technology choice	16
Table 5: Classes and their usage	18

1 Introduction, problem description and approach

The client is a health insurance company that digitized its whole internal document flow. All incoming letters are scanned and stored in a document management system for further processing. For working on the documents, the company uses proprietary software running in a web browser that enables employees to process PDF and TIFF documents.

As TIFF editing is part of the strategic digitalization of company-wide communications, it is crucial for the IT department to achieve independence and flexibility as far as possible while maintaining stability and reliability for tasks such as archiving and audit-proof storage. Setting up this project is triggered by the current implementation, which relies on third-party software that imposes constraints upon the company because of licensing cost, limited functionality and dependence on the third-party provider. It is therefore desirable to improve the situation by switching over to self-maintained solution and realize gains in flexibility and independence.

As part of a university project a team of three people was assigned the task to realize a Java-based tool for TIFF processing/editing ready for use in a web browser. Further goals are the documentation and explanation of technology choice.

The usage of Java is required as per the business requirements of the client and is allowed to be complemented with web technologies if necessary. The approach chosen is the following. First of all, this paper outlines the characteristics of the TIFF format and points out challenges. Based on the knowledge, different technologies that can be leveraged are briefly explained. Moreover, the client's business requirements are documented and further developed using Requirements Engineering. The next chapter documents the exemplary implementation of the developed proof of concept using a selected technology based on defined criteria. Once the requirements are made clear, the actual implementation is described in detail.

2 TIFF – specifications and challenges

TIFF (tag-based file format) was initially developed based on input from scanner manufacturers and software developers and first released in 1986 as Revision 3.0, taking two previous drafts into account.¹ The purpose of this format is, as stated in its final version, “to describe and store raster image data”². A primary goal of TIFF is “to provide a rich environment within which applications can exchange image data”³ in order to take advantage of the varying capabilities of imaging devices. Despite the intention to enhance the TIFF format, the current release – Revision 6.0 – dates back to 1992.

2.1 Characteristics of TIFF

As per the documentation of TIFF, TIFF is an image file format where a file consists of a sequence of 8-bit bytes with at most 2^{32} bytes in length (4 GB):⁴ The file consists of an 8-byte image file header (IFH) and an image file directory (IFD).

The image file header consists of the following items:

Bytes 0-1	“II” or “MM”, indicating the little-endian byte order or big-endian byte order respectively.
Bytes 2-3	“An arbitrary but carefully chosen number (42) that further identifies the file as a TIFF file.” ⁵
Bytes 4-7	The offset of the first IFD (in bytes).

Table 1: TIFF file header parts

¹ Cf. Adobe (1992)

² Adobe (1992)

³ Adobe (1992)

⁴ Cf. Adobe (1992)

⁵ Adobe (1992)

The Image File Directory (IFD) consists of the following parts:

Bytes 0-1	Number of directory entries
Bytes 2-13 and further 12-byte field entries	Directory entries
Bytes x to x+4	Offset of the next IFD (0 if none)

Table 2: TIFF IFD parts

An IFD Entry (12 bytes, in many contexts also referred to as TIFF field) consists of the following parts:

Bytes 0-1	Tag identifying the field.
Bytes 2-3	Field type (Byte, ASCII, Short, Long, Rational, and more).
Bytes 4-7	Number of values, count of the indicated type (Indicates the number of values, which is not the total number of bytes. E.g. a 16-bit word of type SHORT counts one.)
Bytes 8-11	<p>“The Value Offset, the file offset (in bytes) of the Value for the field.”⁶</p> <p>If the value fits into 4 bytes, this contains the value itself. The Type and Count of the field determine if this is possible.</p>

Table 3: TIFF IFD Entry parts

⁶ Adobe (1992)

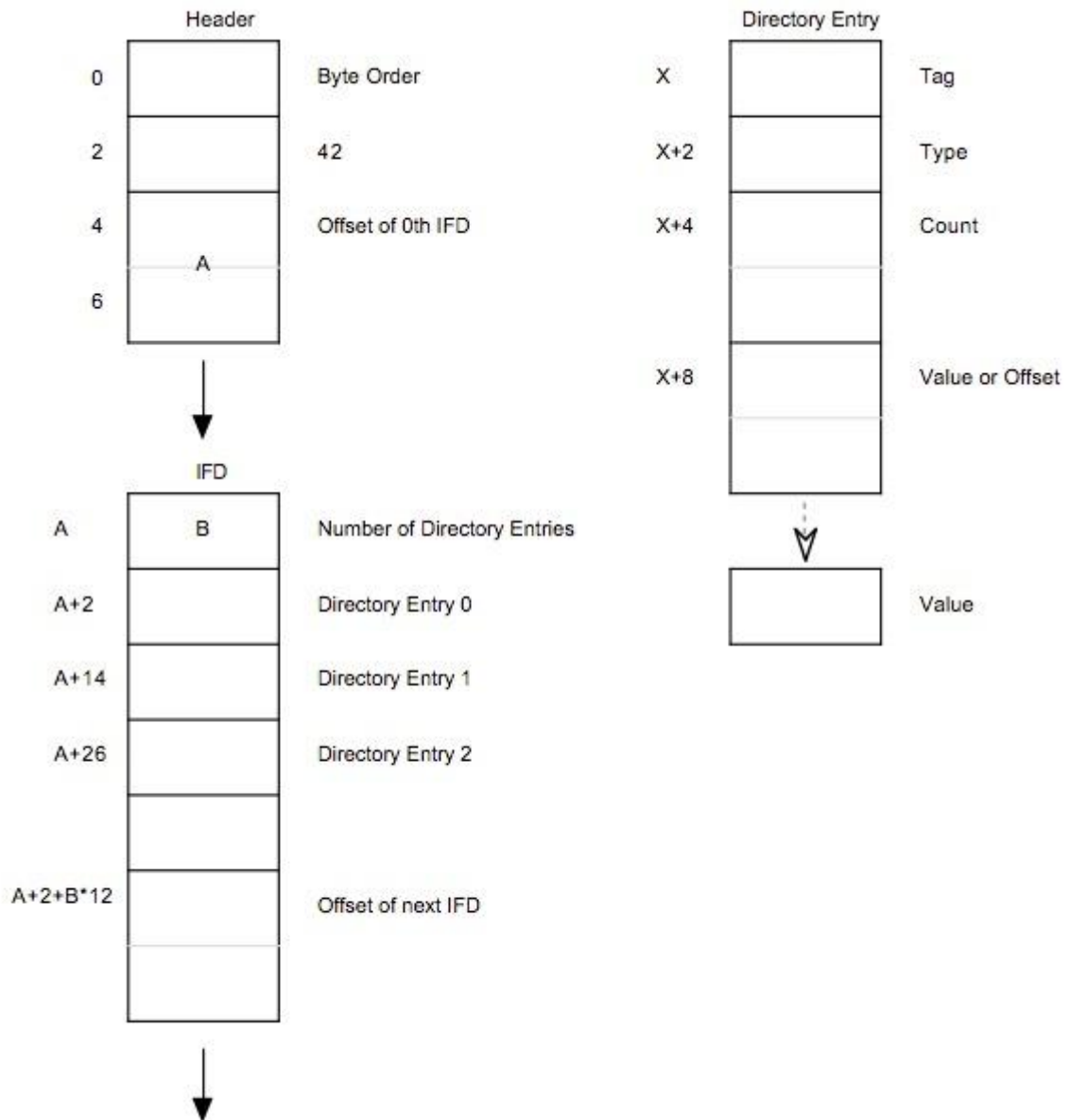


Fig. 1: TIFF illustrated⁷

Fig. 1 illustrates the format. TIFF has four baseline TIFF image types: Bilevel (black and white), grayscale, palette-color and full-color.

2.2 Challenges

There are several challenges that might become relevant when dealing with TIFF files programmatically. The following challenges are true for Baseline TIFF readers:⁸

⁷ Contained in: Adobe (1992)

⁸ Cf. Adobe (1992)

- The use of several IFDs within one TIFF file allows the definition of subfiles. Although a TIFF reader is not required to read more than the first IFD, this possibility has to be considered.
- As per the specification, a field may accept several field types. E.g. ImageWidth accepts SHORT (the usual type) and LONG in case the image has more than 64,000 rows. It is recommended that readers accept Byte, Short and Long for any unsigned integer field.
- TIFF readers must be able to handle non-required fields and must not refuse to read files including optional fields. Examples are Model, DateTime, etc.
- Both big-endian and little-endian byte orders must be read.
- TIFF readers must be able to handle multiple subfiles, even though only the first one (full-resolution image) has to be read.
- TIFF editors must be able to handle subfiles appropriately as well. In case a subfile contains a reduced-resolution version of another subfile, the reduced-resolution image has to be updated (or deleted) upon altering the full-resolution one.
- TIFF editors should furthermore be careful with copying/sustaining fields they cannot cope with, as changes to the image might be incompatible with the fields copied.
- TIFF files may have extra components per pixel, which readers are required to skip over.
- TIFF readers must also skip over unexpected field types.
- TIFF readers must terminate the import process gracefully when pixel data consists of something other than unsigned integers.
- Baseline TIFFs offer several compressions: No compression, PackBits Compression, Modified Huffman Compression

Moreover, there is an extension to TIFF that offers features beyond those that are required for Baseline TIFF readers and editors. This includes CCITT T.4 and T.6 bi-level encoding, LZW compression and JPEG-based compression. Also, the Differencing Predictor can be used to improve the compression ratio. Other features involve Tiled images, CMYK images (separated CMYK image data), Halftone Hints, Associated Alpha Handling, Data Sample Format (describing a variety of data sample formats, such as floating point formats), RGB Image Colorimetry (describing and characterizing RGB image data), YCbCr Images, and CIE L*a*b* images.

Based on that aspects, it is obvious that TIFF is powerful as well as extensible. However, this richness causes complexity too.⁹ Due to the complexity, especially in former times it seemed problematic to develop a TIFF reader/editor that fully supported the TIFF specification: The great flexibility in terms of byte order and compression caused particular difficulties.¹⁰ The IFD holds several chunks of data, as explained above. Those are called tags and contain information about the image. As per the specification, there are more than 70 different tags (e.g. Tag 257 - 101.H stores the ImageLength in Short or Long). While it is not simple to implement the whole specification, there are also private tags that can be defined and might affect the appearance of the image.¹¹ Also, as indicated above, TIFF is able to store multiple images.

⁹ Cf. Murray, J. D./Van Ryper, W. (1996)

¹⁰ Cf. Murray, J. D./Van Ryper, W. (1996)

¹¹ Cf. Proise, J. (1996), p. 321

3 Approaches to TIFF processing

3.1 Native Java (applet)

Java offers native graphic editing (painting) features as part of `java.awt.*` and `java.awt.image.BufferedImage` can be used to handle the image in a readable format. Many image manipulation operations can be realized through those technologies. In addition, there are frameworks that help handle image reading and apply computer vision algorithms. As the native `java.awt.*` features suffice the purpose of the business requirements, the availability of frameworks has minor priority and is naturally low as there is no point in providing native Java functionality in a framework without reasonable improvements. The improvements, however, go beyond the scope of this paper.

3.2 HTML5 and JavaScript technologies

There are various examples¹² that indicate that the use of modern web technologies such as HTML5 and JavaScript enables web developers to create an in-browser environment for comprehensive image processing. Especially the HTML5 tag “canvas” plays an important role. The programming approaches chosen, however, indicate that browser-only processing is not possible or preferred at the time being. Given the complexity of TIFF, it is safe to assume that server-side processing (e.g. through web services) is required to address the challenges of that format. As per the business requirements, the background processing would have to be realized using Java.

3.3 Adobe Flash, Microsoft Silverlight

Although Adobe Flash and Microsoft Silverlight would be a considerable approach, they are excluded as per the given task as they are proprietary technologies.

3.4 Frameworks

In the following section different frameworks for image processing are discussed and investigated for the particular use case of manipulating TIFF files.

¹² Here also confer: <http://demo.leadtools.com/HTML5/> and <http://heidi.deviantart.com/journal/deviantART-Muro-It-s-Time-to-Draw-214232006>

3.4.1 ImageMagick®

ImageMagick® is a software suite to create, edit, compose, or convert bitmap images. Despite many other formats ImageMagick® is often utilized from the command line to process TIFF files.¹³ The software is delivered as a ready-to-run binary or as source code under the Apache 2.0 license. The current version is 6.8.5-9. Interfaces for ImageMagick are available for every major programming language. An interface for the Java programming language is JMagick. It is implemented in the form of Java Native Interface (JNI) into the ImageMagick API. JMagick does not attempt to make the ImageMagick API object-oriented. It is merely a thin interface layer into the ImageMagick API.¹⁴

3.4.2 GraphicsMagick

GraphicsMagick praises itself as the swiss army knife of image processing.¹⁵ It is applicable for the defined situation, because of the support for huge images. GraphicsMagick may be used to create, resize, rotate, sharpen, color reduce, or add special effects to an image and save the result in the same or differing image format. Image processing operations are available from the command line, as well as through C, C++, Lua, Perl, PHP, Python, Tcl, or Ruby. The absence of Java excludes GraphicsMagick from the preferred choices.

3.4.3 Netpbm

Netpbm is a toolkit for manipulation and conversion of graphic images between a variety of different formats. 300 separate tools are in the package including converters for about 100 graphics formats. Exemplary features comprise shrinking, cutting the top half off, making a mirror image, creating a sequence of interspersing fading images.¹⁶ The copyright for the source code belongs to the specific author of the code. All of these public licenses are "open source" licenses as defined by SourceForge. One can generally find an offer of a copyright license within the source code files. GPL, BSD, MIT, and BSD licenses are among those offered. The popularity of this framework is mostly historical. The simplification and reduction of different used file formats caused an inevitable decline of converter importance. Therefore we will not consider this framework for this project.

¹³ Cf. ImageMagick (w. y.)

¹⁴ Cf. JMagick (w. y.)

¹⁵ Cf. GraphicsMagick (w. y.)

¹⁶ Cf. Netpbm (w. y.)

3.4.4 VIPS

Another free image processing system is VIPS. It comes with two main parts: libvips is the image-processing library and nip2 is the graphical user-interface. VIPS advantages over other systems lie in the efficient usage of multiple CPUs. It is particularly valuable when the file size is larger than the available RAM.¹⁷ Libvips can be used from C, C++, the command-line, Python, Ruby, and JavaScript. Additionally it is licensed under the LGPL and nip2 is licensed under the GPL.

3.4.5 Apache Commons Imaging

The previously known as Apache Commons Sanselan library reads and writes a variety of image formats. In addition, it includes fast parsing of image information and metadata.¹⁸ The library is pure Java and therefore perfectly portable and the best fit for our purpose. Even though it is slower than other frameworks the benefits of easy usage, clean interface, network-friendliness are predominant.¹⁹ The open source project is available under the Apache License (non-viral open source license).

¹⁷ VIPS (2013)

¹⁸ Apache Commons (w. y. a)

¹⁹ Apache Commons (w. y. b)

4 Exemplary implementation

This section describes the requirements and their implementation as well as the technology choice. Based on the requirements, the technology is chosen and the implementation is explained afterwards.

4.1 Requirements Engineering

Requirements Engineering is a standardized way of defining and managing requirements. The following structure is based on IEEE (1998) and adapted as appropriate for the scope of this paper.

4.1.1 Purpose, scope and overview

The purpose of this prototype is to demonstrate the abilities for TIFF editing with freely available Java-based frameworks and native Java methods. The use case defined by the client entails various functions such as redaction, stamping, commenting, cutting, and saving of TIFF images for users with no particular IT knowledge. In addition, the software has to fit in an existing environment that relies on a web-based Java application and meet the technical constraints of what employees are allowed use.

4.1.2 Product perspective

The product perspective is hard to be determined in this early stage. The product will interact with the client's database system, which is not part of the prototype and therefore only set up as a similarly working MySQL database.

4.1.3 Product functions (business requirements)

The following business requirements were determined within the given tasks and individual interviews with the client:

- **Loading:** The software must have the ability to load images from a hard disk.
- **Redaction:** The need to make documents anonymous is crucial in the business environment of an insurance company. Redaction of client names, credit card numbers, and disclosure dates enables effortless and carefree distribution of documents and ensures the compliance with the law, governance regulations, and internal compliance rules. Once redaction was applied, it should not be necessary to reverse this action.
- **Stamping:** There are many reasons to stamp TIFF documents, but the main one would be to indicate the date the document was received (or scanned).

The stamp includes the name of the employee who has processed the document as well as the date this happened.

- **Post-it like Commenting:** Employees should be enabled to write comments on the image in a way that can be compared with the analog post-it.
- **Copying:** Parts of the TIFF image can be copied into the clipboard and pasted into other documents.
- **[Optional] Undo an action:** Whenever the user has carried out an action that was undesired, it can be reversed with an undo function.
- **[Optional] Highlighting:** Highlight parts of the image by adding a transparent, colored layer.
- **Saving:** The changes applied to the document can be saved into a file on the local hard disk. There is no server or network volume interaction developed within this prototype.

General data related to the image (author, last change, other versions, image name, MD5, location on server) are also stored in a database, which should be updated accordingly.

4.1.4 User characteristics

People using the functionality of this prototype typically have particular expertise in insurance topics and customer interaction. They are required to manage a certain number of transactions and documents a day and therefore are reliant on a fast and easy-to-use software that runs in their common environment. Those people cannot necessarily take advantage of particular IT expertise and cannot be expected to manage or actually recognize unintended software behaviors.

4.1.5 Constraints

The environment available at the client is standardized and each system must meet company-wide IT guidelines. The current software is a JSP-based web application running on WebSphere 6 and Java 6.3. The standard web browser installed on all computers is Internet Explorer (IE) 9 and cannot be substituted or complemented by additional browsers. The application has to be developed in a similar environment and might make use of HTML5, CSS3 and JavaScript, provided IE 9 support is ensured. Also a Java Applet is a valid approach.

4.1.6 Apportioning of requirements

There are various features that are not requested to be part of this prototype, but might become relevant in the future. A choice of those features is described in the following paragraphs.

- **Reduction of image file size:** TIFF offers limited capabilities to apply compression mechanisms, which is why it is generally possible to reduce the image size. Prior to implementation, it would be important to assess the implications for further editing and other technical aspects.
- **Reduction of image size:** For some reasons, such as showing thumbnails, it might become desirable to reduce the image size.
- **Painting geometric forms such as lines, circles, rectangles, triangles, etc.** could be added and might be desirable to improve the ability to highlight parts of an image.

4.1.7 Specification – External interfaces, product and performance requirements

This section specifies the technical realization of the business requirements mentioned above. The detailed requirements are presented in the following way:

- *ID #xxx, Title: Description*
 - *Sequence of operations:*
 - *Step 1*
 - *Step 2*
 - *Step n*
 - *Validity checks: ...*
 - *Relation to other data records: ...*
 - *Data conversion: ...*
- ID #001, Loading images: The user can load images from the hard disk.
 - Sequence of operations:
 - The user opens the software (assumption: the prototype does not involve authentication) and clicks “load” to select a file on the hard disk
 - Load the image’s meta data from the image database in order to be able to refer to that later on
 - The image is shown in a dialog of the software

- The user is presented the palette of actions available for use (see business requirements)
 - The user can apply one or more actions to the image
 - Validity checks: Prior to opening the TIFF file, a validity check takes place to make sure the software is able to work with the given format of TIFF (see Challenges for further information). Provided the check was successful, the user will be presented the image.
 - Relation to other data records: Although not part of the prototype, access rights management might be applied here later on
 - Data conversion: Apache Commons Imaging automatically creates a BufferedImage for further processing
- ID #002, Redaction: The user can hide information by adding black rectangles, whose size can be defined variably.
 - Sequence of operations:
 - Follow the operations described in #001
 - Click “redaction”-symbol
 - Indicate the area on the image that has to be redacted
 - Check if the result is appropriate
 - Validity checks: -
 - Relation to other data records: -
 - Data conversion: -
- ID #003, Stamping: The user can add a stamp on the document that indicates for example the date it was received and employee it was processed by. The Date can be defined manually.
 - Sequence of operations:
 - Follow the operations described in #001
 - Click “stamp”-symbol
 - Define a caption that will be stamped on the document (e.g. “RECEIVED”, “PAID”)
 - Define a date that will be stamped on the document
 - Check if the result is appropriate
 - Validity checks: -
 - Relation to other data records: -
 - Data conversion: -

- ID #004, Post-it like commenting: The user can write a comment, which will be written on the image in a highlighted format.
 - Sequence of operations:
 - Follow the operations described in #001
 - Click “comment”-symbol
 - Write comment in a text field
 - Indicate the position it should be placed at on the image
 - Check if the result is appropriate
 - Validity checks: -
 - Relation to other data records: -
 - Data conversion: -
- ID #005, Copying: The user can select a specific area of the image and copy it to the clipboard.
 - Sequence of operations:
 - Follow the operations described in #001
 - Click “Copy”-symbol
 - Select area on the image
 - Go on and copy the result into a different software
 - Validity checks: -
 - Relation to other data records: -
 - Data conversion: -
- #006, Saving: Once the processing is finished, the user can save the TIFF image as a new file to the hard disk.
 - Sequence of operations
 - Follow the operations described in #001
 - Follow one or many operations described in #002 to #005
 - Click the “save”-button and choose the location the image should be saved to.
 - Save a new data record to the image database that is related to the old one.
 - Validity checks: -
 - Relation to other data records: The processed file should be related to the original file in the database using a foreign key (FK).
 - Data conversion: -

4.1.8 Specification – Component requirements

- Java: JEE 1.6.3
- Database: MySQL Version 5.5.27
- Permission to run privately signed Java Applets within Internet Explorer 9

4.1.9 Specification – Logical database requirements

The database serves as centralized storage for general image meta information and integrity checks leveraging MD5 check sums. See Fig. 2 for the database schema.

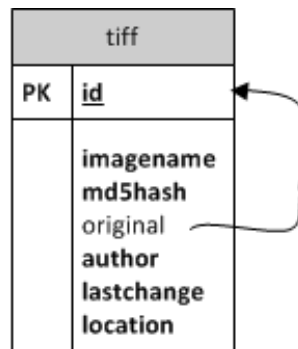


Fig. 2: Database schema

4.2 Application of decision criteria for technology choice

Criterion / Technology	Java Applet <i>A Java Applet can be coded entirely using Java source code and is platform independent available.</i>	HTML5 and JavaScript in conjunction with Java-based backend <i>The client-side editing would be realized using web technologies. As indicated above, server-side processing is needed to support certain tasks (e.g. reading, saving, database interaction)</i>
Reliability (audit-proof)	High – As secure as Java Applets are, code manipulation is unlikely.	Medium – Basically it cannot be made sure that client-side code is unchanged and requests to the server are clean.
Compliance with technical requirements (Internet Explorer 9, Java 1.6)	High	High – Internet Explorer 9 supports the HTML5 “canvas” tag ²⁰ and JavaScript usage can be tailored to possible IE9 particularities.
Ease of Maintainability	High – Only Java skill is required for	Low – The combination of HTML5, JavaScript, possibly CSS for

²⁰ See <http://caniuse.com/#search=canvas>

	development.	styling, and Java makes it more difficult for development.
Readiness for individual requirements (ease of development)	High – all business requirements can be met.	High – all business requirements can be met.
Skill availability	Java skill is available as the business requirements indicate.	Java skill is available as per the given task. Web development skills are unknown.
Availability of frameworks	All business requirements can be met with reasonable effort.	There are hardly Open Source solutions available.

Table 4: Decision criteria for technology choice

4.3 Exemplary TIFF processing tool

The proof of concept is developed using native Java and Apache Commons Imaging. As indicated above, the native Java features for image manipulation suffice the requirements identified. The main challenge therefore is to read all kinds of TIFF reliably and write manipulations back into a TIFF accordingly. Apache Commons Imaging is the most promising framework for this task, as Apache is a well-respected institution for open source projects and the functionality provided has proven perfect in initial tests. This chapter describes the software design and its implementation.

4.3.1 Software design

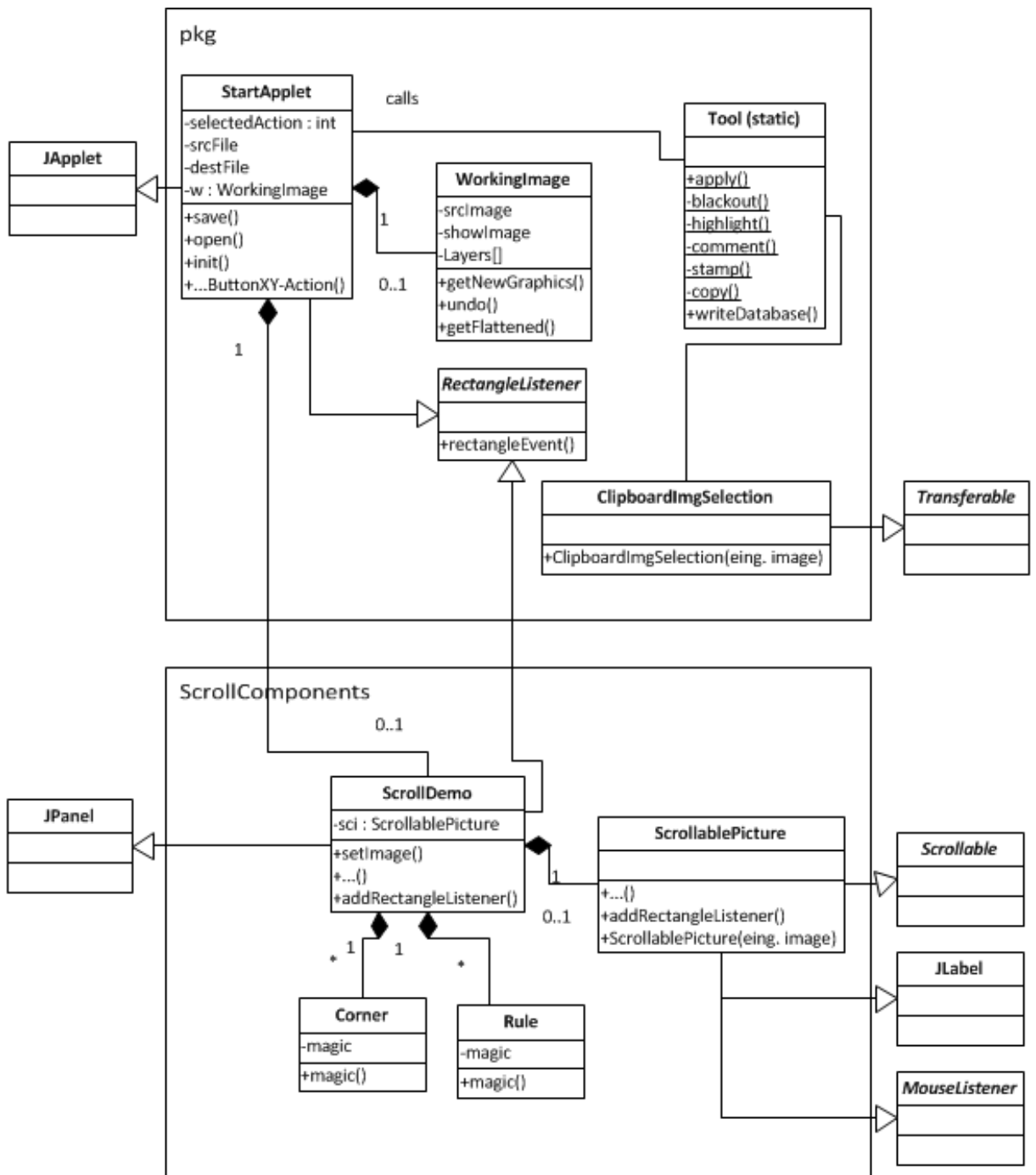


Fig. 3: UML class diagram

Fig. 3 illustrates the implementation of the prototype. Further documentation is available within the source code itself (available digitally) through comprehensive comments.

4.3.2 Software implementation

The software consists of nine java classes. The following table describes them and their corresponding purposes.

Name	Purpose
Corner.java	Corner.java is used by ScrollDemo.java.
Rule.java	Rule.java is used by ScrollDemo.java.
ScrollDemo.java	ScrollDemo.java is source code provided by Oracle.
ScrollablePicture.java	ScrollablePicture.java is used by ScrollDemo.java. It reuses the scrolldemo app provided by Oracle and adjusts it in some parts to fit the program design.
ClipboardImgSelection.java	The content pushed into the clipboard needs to be implemented as transferable. The algorithm is copied from StackOverflow and nothing except the constructor is called by the authors' code.
RectangleListener.java	This interface needs to be implemented to listen to the selection rectangle events.
StartApplet.java	This is the graphical user interface and the place where the business logic happens. All actions are implemented here and the whole applet is initialized.
Tool.java	This is the toolbox, it carries out the required operations with the working image. It can be considered as a utility for the program.
WorkingImage.java	This class contains some handling methods for convenient manipulation of the central image and supports undo and layers.

Table 5: Classes and their usage

5 Conclusion

As part of a university project a team of three people had the goal to realize TIFF processing/editing within a web tool using Java, aided by open source frameworks. The usage of Java was required as per the business requirements of the client and was allowed to be complemented by web technologies if necessary.

The first value-add of this paper is characterized by a comprehensive introduction to TIFF and corresponding challenges. Afterwards, several frameworks were introduced in order to give an overview of the market, always bearing the overall business case in mind. Once the market overview was provided, the authors holistically analyzed the client's business requirements and split them into multiple components while following the standards of Requirement Engineering defined by IEEE.

Considering all business requirements, it turned out that even though several frameworks are available for TIFF processing, the most appropriate solution for the client's business case is a software neatly developed in native Java while leveraging Apache Commons Imaging for image reading and writing. On the one hand this was due to the business constraints given by the client, and on the other hand required by technological limitations of the discussed (partly open source) frameworks. The final decision was made based on a comprehensive set of criteria applied to the technologies available.

The software that was realized within a proof of concept is ready for use on every productively deployed platform within the company and in addition is potentially ready for future application areas in the era of mobile computing. Object-oriented programming, a sustainable structure and complete documentation enable the client to transform the proof of concept into productively usable software with little effort.

Appendix: Prototype Software Requirement Specification (SRS)

Table of Contents	
1.	Introduction
1.1	Purpose
1.2	Scope
1.3	Definitions, acronyms, and abbreviations
1.4	References
1.5	Overview
2.	Overall description
2.1	Product perspective
2.2	Product functions
2.3	User characteristics
2.4	Constraints
2.5	Assumptions and dependencies
3.	Specific requirements (See 5.3.1 through 5.3.8 for explanations of possible specific requirements. See also Annex A for several different ways of organizing this section of the SRS.)
	Appendixes
	Index

Fig. 4: Prototype SRS outline²¹

²¹ IEEE (1998), p.11

List of references

List of literature

- IEEE (1998): IEEE Recommended Practice for Software Requirements Specifications, IEEE Std 830-1998, IEEE Computer Society, p. 1-39
- Prosise, J. (1996): A Look Inside Bitmap Files, in: PC Magazine, Vol. 15, December 3, p. 321-323

Internet sources

- Adobe (1992): TIFF, Revision 6.0, Final, <http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf>, Retrieval: 04.06.2013
- Apache Commons (w. y. a): Apache Commons Imaging, <http://commons.apache.org/proper/commons-imaging/>, Retrieval: 05.06.2013
- Apache Commons (w. y. b): Why Apache Commons Imaging, <http://commons.apache.org/proper/commons-imaging/whyimaging.html>, Retrieval: 05.06.2013
- GraphicsMagick (w. y.): GraphicsMagick, <http://www.graphicsmagick.org>, Retrieval: 05.06.2013
- ImageMagick (w. y.): Convert, Edit, And Compose Images, <http://www.imagemagick.org/script/index.php>, Retrieval: 05.06.2013
- JMagick (w. y.): SourceForge JMagick, http://sourceforge.net/apps/mediawiki/jmagick/index.php?title=Main_Page, Retrieval: 05.06.2013
- Murray, J. D./Van Ryper, W. (1996): Encyclopedia of Graphics File Formats, 2nd Edition, <http://www.fileformat.info/format/tiff/egff.htm>, Retrieved: 15.06.2013
- Netpbm (w. y.): SourceForge Netpbm, <http://netpbm.sourceforge.net>, Retrieval: 05.06.2013
- VIPS (2013): VIPS, <http://www.vips.ecs.soton.ac.uk/index.php?title=VIPS>, Retrieval: 06.06.2013

Marktanalyse und Vergleich clientseitiger Open Source Frameworks zum Bau browserbasierter Desktop-Anwendungen

Schriftliche Ausarbeitung
im Rahmen der Lehrveranstaltung „Open Source Seminar“
Kompetenzzentrum Open Source (KOS)

vorgelegt von

Meuth, Isabelle; Mielke, Janina;
Thöne, Jan-Philipp; Tuchscherer, Lena
am 01.07.2013

Fakultät Wirtschaft
Studiengang Wirtschaftsinformatik
WWI-2010-I

Inhaltsverzeichnis

Abkürzungsverzeichnis	III
Abbildungsverzeichnis	IV
Tabellenverzeichnis	IV
1 Einleitung	1
1.1 Zielsetzung	1
1.2 Abgrenzung	2
1.3 Vorgehen	2
2 Definitionen und Erklärungen.....	3
2.1 Grundbegriffe	3
2.2 Technologien für clientseitige Entwicklung.....	4
2.3 Architekturmodelle für Webdarstellungen	5
2.3.1 MVC-Pattern	5
2.3.2 MVVM-Pattern	7
3 Überblick und Kategorisierung von Frameworks am Markt	9
3.1 Übersicht von existierenden Frameworks	9
3.1.1 UI Komponentenframeworks	11
3.1.2 Anwendungserstellungsframeworks	13
3.2 Überblick über die am Markt befindlichen Frameworks	14
3.3 Eingrenzung marktrelevanter Frameworks	15
4 Erarbeitung eines Kriterienkatalogs zur Bewertung von Frameworks	18
4.1 Rechtfertigung und Beschreibung der Vorgehensweise	18
4.2 Beschreibung Bewertungsmodell	19
4.3 Erarbeitung der Evaluations-Kriterien	20
4.3.1 Findung, Gruppierung und Bewertung der Kriterien	20
4.3.2 Vorstellung des Kriterienkatalogs	25
5 Anwendung des Kriterienkatalogs	27
6 Fazit	33
Anhang.....	34
Quellenverzeichnisse	42

Abkürzungsverzeichnis

AJAX	A synchronous J avaScript and X ML
(G)UI	(G raphical) U ser Interface
HTML	H ypertext M arkup L angugae
HTTP	H ypertext T ransfer P rotocol
JS	J ava S cript
MFC	M icrosoft F oundation C lasses
MVC	M odel V iew C ontroller
MVVM	M odel V iew V iew M odel

Abbildungsverzeichnis

Abbildung 1: MVC-Pattern	5
Abbildung 3: MVVM-Pattern	7
Abbildung 4: Beispiel Struktur des UI Komponentenframeworks Twitter Bootstrap	11
Abbildung 5: Beispiel für Komponenten von Twitter Bootstrap	12
Abbildung 6: Überblick an Open Source Frameworks auf dem Markt	14
Abbildung 7: Vergleich marktrelevanter Frameworks mit Google Trends	17

Tabellenverzeichnis

Tabelle 1: Vergleich Anwendungs- und Komponentenframework	10
Tabelle 2: Einstellung für die Google Trends Analyse	16
Tabelle 3: Kriterienkatalog mit Gewichtungen	25
Tabelle 4: : Beispielhafter Vergleich zweier Frameworks	26
Tabelle 5: Auswertung nach Kriterienkatalog – Twitter Bootstrap	29
Tabelle 6: Anwendung des Kriterienkatalogs - Dojo	30
Tabelle 7: Gegenüberstellung Twitter Bootstrap und Dojo	31

1 Einleitung

Dienstleistungen werden immer öfter über webbasierte Anwendungen bereitgestellt und sind somit fester Bestandteil von vielen Geschäftsprozessen und –Modellen von Unternehmen aller Art. Diese webbasierten Anwendungen haben sich im Laufe der Jahre immer stärker entwickelt und verändert.

Durch den Einsatz von Open Source, das heißt öffentlich zugängliche Quelltexte, die frei modifiziert und weiterverarbeitet werden dürfen¹, konnten Frameworks das heißt Programmiergerüste geschaffen werden, die bei der Erstellung von webbasierten Anwendungen Unterstützung leisten bringen. Hier unterscheidet man zwischen server- und clientseitigen Technologien. Während serverseitig die Anwendung noch auf dem Server ausgeführt wird und lediglich der Inhalt über den Client dargestellt wird, wird bei der clientseitigen Technologie bereits die Anwendung schon vom Client gesteuert, während der Server nur noch die fachliche Logik betreut.² Der derzeitige Trend tendiert zu clientseitigen Technologien, durch die auch sogenannte Desktop-Browseranwendungen erstellt werden können. Aber auch mobile Browseranwendungen können mit dieser Technologie bedient werden.

JavaScript gilt als Basistechnologie für clientseitige Anwendungsframeworks. Die Menge an Frameworks ist noch schwer zu überschauen und es fällt schwer noch Unterschiede auf den ersten Blick zu erkennen.

Daraus ergibt sich eine Problemstellung, der sich in dieser Seminararbeit gewidmet werden soll: Der Entwicklung eines Kataloges, der die Bewertung von Frameworks anhand von Kriterien ermöglicht. Durch einen solchen Katalog soll erreicht werden, dass im Falle der Entwicklung von clientseitigen Desktop-Anwendungen schnellstmöglich die Entscheidung für ein passendes Framework gefällt werden kann, indem die entsprechenden Kriterien der zur Verfügung stehenden Frameworks miteinander verglichen und auf ihre Tauglichkeit hin bewertet werden. Somit kann durch diese Arbeit Entwicklern von Desktop-Browseranwendungen der komplizierte Einstiegspunkt in seine Arbeit erleichtert werden.

1.1 Zielsetzung

Ziel der Arbeit ist eine Marktanalyse derzeitiger Frameworks. Dazu soll eine Bewertungsgrundlage zur Auswahl eines passenden Frameworks basierend auf unterschiedliche Kriterien geschaffen werden, was einem Bewertungskatalog gleicht. Vorhandene Frameworks müssen erarbeitet und kategorisiert werden, daraufhin wird ein Kriterienkatalog erstellt, der

¹ Vgl. BpB (2013)

² Vgl. Hitz, M. (2013)

die Bewertung von Frameworks vorantreibt. Abschließend soll anhand eines exemplarischen Vergleiches der Vergleich zweier Frameworks erörtert werden.

1.2 Abgrenzung

Im Rahmen dieser Seminararbeit wird sich lediglich mit der clientseitigen Technologie beschäftigt, die zur Erstellung von Desktop-Browseranwendungen führt. Die Thematik der mobilen Browseranwendungen wird in einer weiteren Seminararbeit abgehandelt und ist somit nicht Grundlage dieser Arbeit.

1.3 Vorgehen

Die Seminararbeit beinhaltet theoretische und praktische Aspekte. Zum Einstieg wird unter dem Kapitel Definitionen und Erklärungen eine theoretische Grundlage geschaffen. Neben grundsätzlichen Definitionen werden Technologien und Programmiermodelle beschrieben. Darauf folgen ein Überblick und die Kategorisierung verschiedener Frameworks am Markt, gefolgt von der Erarbeitung von Bewertungskriterien und deren Zusammenhänge, die am Ende in einem Kriterienkatalog münden. Ein exemplarischer Vergleich und ein abschließendes Fazit schließen die Seminararbeit ab.

2 Definitionen und Erklärungen

Die nachfolgenden Definitionen und Erklärungen sollen als theoretische Grundlage für die spätere Erarbeitung eines Kriterienkatalogs für Frameworks gelten.

2.1 Grundbegriffe

Clientseitige Technologien

Clientseitige Technologien laufen nicht auf einem Server sondern auf den Rechnern ihrer Nutzer ab. Sie kommunizieren zum Server und holen über diesen Informationen ab, besitzen aber keine Abhängigkeit vom Server und können dementsprechend auch offline auf dem Browser laufen. Diese Technologien können übergebene Daten vom Server lokal verwalten.

JavaScript und Ajax gehören zu den bekanntesten Vertretern der clientseitigen Technologien.

Vorteile von Clientseitigen Technologien sind eine effizientere Kommunikation und konventionelle Anwendungsentwicklung, während die Nachteile bei der Sicherheit und fehlenden alternativen Endgeräten liegen. Browserbasierte Desktop-Anwendungen sind Teil der clientseitigen Technologien.

Desktop Browser-Anwendungen

Webanwendungen sind Anwendungsprogramme, die in Webbrowsern ihrer Benutzer ablaufen. Sie werden größtenteils auf einem Webserver gespeichert und ausgeführt und über das Internet oder ein Protokoll wie http abgerufen werden.³

Browserbasierte Anwendungen gewinnen immer mehr Benutzer und dazu auch an Bedeutung, die Vorteile von Webanwendungen, insbesondere die Plattformunabhängigkeit und die Einfachheit der Benutzung gegenüber Desktopanwendungen, sind vielfältig und dienen der Popularität. Oftmals gibt es allerdings noch Bedenken bei der Sicherheit von browserbasierten Anwendungen. Open Source Programme bieten allerdings in der Regel eine hohe Sicherheit, da der Quelltext allgemein zugänglich ist und Lücken somit schneller gefunden und behoben werden können. Updates, insbesondere Sicherheitsupdates, können schnell und einfach verbreitet und durchgeführt werden.⁴

Ein weiterer positiver Aspekt von Webanwendungen ist die Flexibilität und die damit verbundenen geringeren Kosten von Webanwendungen. Webanwendungen können als browserba-

³ Vgl. Wikipedia (2013a)

⁴ Vgl. Media Engineering (2013)

sierte Anwendungen von jedem Arbeitsplatz aufgerufen werden und erlauben einen flexibleren Arbeitsalltag. Desktopanwendungen, die lizenziert gekauft werden müssen und nur an bestimmten Arbeitsplätzen verwendet werden können, stehen daher in besonderem Nachteil gegenüber browserbasierten Anwendungen. Auch das Arbeiten über eine Cloud, die ermöglicht, dass man nur die wirkliche Benutzerzeit für die Anwendung zahlt, gilt als Kostenfortschritt. Open Source Anwendungen lassen sich insbesondere für die individuellen Wünsche der Benutzer frei gestalten. Plugins und Erweiterungen können jederzeit in die offenen Quellcodes eingebaut werden, die neue, individuell nutzbare Funktionen für Webanwendungen bringen.⁵

Auch der Support für browserbasierte Anwendungen ist einfacher, da er von Remote aus einfach per Login durchzuführen ist.

Neben Desktopanwendungen profitieren natürlich auch Mobile Geräte wie Smartphones und Tablet-PCs in besonderer Weise von solchen Webanwendungen.

2.2 Technologien für clientseitige Entwicklung

Ein Programm besteht aus definierten Sequenzen von Befehlen an den Prozessor eines Computers mit der Intention Daten zu transformieren. Die strukturierte, einheitliche Art und Weise in der die Befehle geschrieben werden, wird Programmiersprache genannt. Programmiersprachen unterscheiden sich in Syntax und Semantik. Ersteres beschreibt die logische und formal definierte Abfolge der Elemente, vergleichbar mit der Grammatik. Die Semantik hingegen weist einzelnen Begriffen eine Bedeutung zur zielgerechten Benutzung zu.⁶

Die Programmiersprache JavaScript, Kurzform JS, ermöglicht die Adaption von Inhalten und die Analyse von Nutzerinteraktion in Web-Browsern. JavaScript kann auf viele Objekte, wie den Web-Browser selbst zugreifen, besitzt selbst jedoch nur wenige Objekte.⁷ Die textbasierte Sprache HTML5 dient vor allem der Strukturierung von Bildern, Texten und Hyperlinks, genauso wie dynamische Grafiken, Video und Audiodateien. HTML5 soll interaktive Applikationen erstellen und erweitert die Funktionen zur Formulardatenerstellung mit der Intention die Validierung durch JavaScript abzuschaffen.⁸

Die Vorteile dieser beider Strukturierungsmodelle und Sprachen werden in dem Konzept für dynamische Webseiten „AJAX“ vereint. AJAX steht für „Asynchronous JavaScript and XML“ (Extensible Markup Language). Wie der Name besagt, besteht das Konzept aus einer asyn-

⁵Vgl. Media Engineering (2013)

⁶Vgl. Henning, P. / Vogelsang, H. (2007), S. 24

⁷Vgl. Steyer, R. (2010), S. 36

⁸Vgl. Lawson, B./ Shar, R. (2011), S.1

chronen Kommunikation zwischen Server und Browser. Basierend auf HTML zur dynamischen Darstellung und JavaScript zur Verbindung der Anzeigeobjekte, ermöglicht es sowohl ein gleichzeitiges Darstellen einer HTML Seite und eingegebenen Benutzerdaten als auch die Vornahme von Veränderungen auf der Seite ohne diese erneut aktualisieren zu müssen.⁹

Anwendungen die auf AJAX Basis funktionieren, haben den Anschein komplett auf dem Desktop des Users zu laufen, basieren jedoch auf HTTP- Anfragen und den Request-Response Prinzip. Damit keine längeren Wartezeiten entstehen durch verzögerte Server-Response, sind die Anwendungskomponenten verteilt. Wenn eine Komponente in einem Formular geändert wird, so wird ausschließlich ein Request für den entsprechenden Seitenteil an den Server gestellt. Der Web-Browser baut diesen Teil mit JavaScript neu und so gut wie verzögerungsfrei auf. Für Aufgaben, die der Client selbst erfüllen kann, sind keine Requests notwendig.¹⁰

Für Webanwendungen, vor allem im Business-to-Business Bereich, ist diese Technologie von großer Bedeutung.

2.3 Architekturmodelle für Webdarstellungen

2.3.1 MVC-Pattern

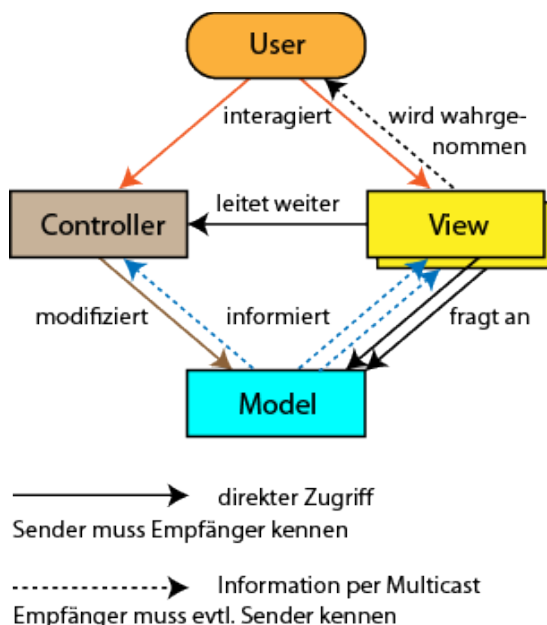


Abbildung 1: MVC-Pattern¹¹

⁹ Vgl. IT Wissen (2013a)

¹⁰ Vgl. Steyer, R. (2010), S.101

¹¹ Vgl. HS-Augsburg (2011)

Das Model-View-Controller Programmiermodell ist ein Muster, das die Module Model, View und Controller genauer definiert:¹²

Im Modul Model werden hier die darzustellenden Daten, die Geschäftslogik und der Datenzugriff geregelt und klassische Objekte eines objektorientierten Programms oder Datenbankzugriffs könnten sich hier befinden. Das Model ist von den anderen Modulen unabhängig.

Das View Modul (auch Präsentationsschicht genannt) hingegen ist für die Darstellung der Daten aus dem Model zuständig und beinhaltet alle Ansichten eines Programms, die die Daten eines Models anzeigen und unabhängig von den Daten implementiert werden. Die Präsentationsschicht ist jedoch nicht für die Weiterverarbeitung der vom Benutzer übergebenen Daten zuständig.

Die Kontrollmöglichkeiten, die dem Benutzer komplett unabhängig von den anderen Bestandteilen, die Steuerung des Programms ermöglichen, befinden sich im Controller Modul. Dieses verwaltet eine oder mehrere Präsentationen und ist für die Entgegennahme und Auswertung der Benutzeraktionen zuständig. Die Controller Schicht hat außerdem die Aufgabe, die vorhandenen Daten anhand der Benutzeraktion entsprechend zu ändern oder zu manipulieren.

Dadurch, dass die Module miteinander kommunizieren müssen, entstehen Abhängigkeiten, die die Komplexität und Fehleranfälligkeit erhöhen. Somit sollten Abhängigkeiten vermieden werden um Fehler vorzubeugen.

Die Module können auf unterschiedliche Weise miteinander kommunizieren. Zum einen können sie Unicast-Nachrichten senden, in denen ein Modul die Methoden eines anderen direkt aufruft. Zum anderen können sie alle Module, die als Nachrichtenempfänger geeignet sind, in Form von Multicast-Nachrichten mit Hilfe von Signalen informieren. Der Sender informiert hierbei den Empfänger, um ihm mitzuteilen, dass sich etwas geändert hat und der Empfänger muss sich somit die für ihn wichtigen Informationen vom Sender holen. Der Sender schickt dann in einer Nachricht alle detaillierten Informationen mit, sodass der Empfänger nicht mehr auf den Sender zugreifen muss.¹³

¹² Vgl. Lott, J./Patterson, D. (2007), S.46

¹³ Vgl. HS-Augsburg (2011)

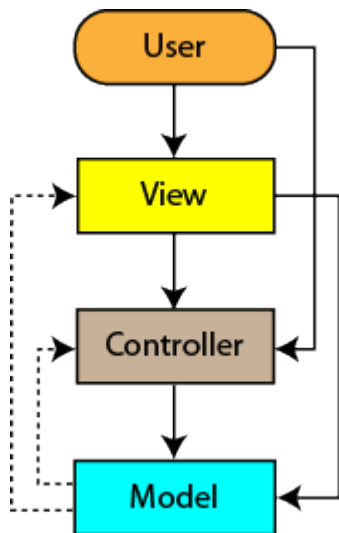


Abbildung 2: MVC Pattern

Wie in Abbildung 2 zu sehen, kann man das Modell in verschiedenen Schichten anordnen: **In der untersten Schicht ist das Modell, darüber liegt der Controller und abschließend die View.**

Dieses Schichtenpattern ermöglicht nur den höheren Schichten Zugriff auf die tieferen, jedoch nicht umgekehrt. Die View darf also sowohl auf Controller als auch auf Model zugreifen. Der Controller darf auf das Model zugreifen und dieses auf kein anderes Modul.¹⁴

Diese Einteilung ist insofern sinnvoll, dass die unteren Schichten keine Informationen über die Verarbeitung aus den oberen Schichten benötigen. So ist es für den Controller, der die Benutzereingaben verarbeitet z.B. nicht relevant, welche Views das Modell darstellt. Eine View hingegen muss die unteren Schichten kennen, um die Daten sinnvoll zu visualisieren.

2.3.2 MVVM-Pattern

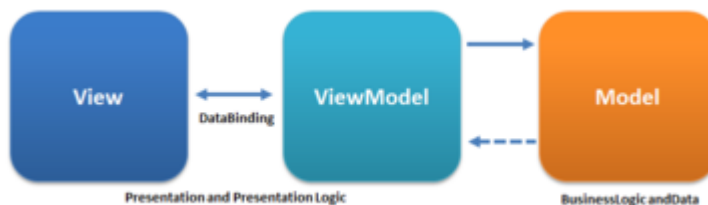


Abbildung 3: MVVM-Pattern¹⁵

Das Model View ViewModel (MVVM) Programmiermodell stammt vom MVC Modell und dient der Trennung von Markup und Logik des User Interfaces. Da sich die IT schnell wandelt, zielt es auf moderne UI-Plattformen wie z.B. Silverlight und HTML 5 ab. MVVM will hier die Rolle der UI-Designer und die der Entwickler trennen, wodurch Anwendungsschichten durch verschiedene Arbeitsgruppen entwickelt werden können. Die Designer sollen hier ein Augenmerk auf die User Experience legen und die Entwickler auf die UI- und Geschäftslogik.

¹⁴ Vgl. Langner, T./Reiberg, D. (2006), S.488

¹⁵ Vgl. Wikipedia (2013b)

Die im MVVM beinhalteten Komponenten bauen sich wie folgt auf:¹⁶

Die Model-Schicht wird auch Datenzugriffsschicht genannt und ähnelt der des MVC Modells. Sie ist auch hier für die Inhalte, die dem Benutzer angezeigt werden zuständig und muss diese in geeigneter Form manipulieren. Sie benachrichtigt andere Schichten über Datenänderungen und führt eine Validierung der Benutzereingaben durch.

Die Präsentationsschicht (View) beinhaltet alle durch die GUI angezeigten Elemente und bindet sich an Eigenschaften des ViewModel. Dies dient der Darstellung und der Manipulation von Inhalten und der Weiterleitung von Benutzereingaben.

Die UI-Logik ist Grundlage der ViewModel-Schicht, welche als Bindeglied zwischen View und Model agiert. Sie tauscht Informationen mit dem Model aus, d.h. sie ruft Methoden oder Dienste auf und stellt der View öffentliche Eigenschaften und Befehle zur Verfügung, welche von der View an Steuerelemente gebunden werden, um Inhalte und UI-Ereignisse auszugeben und weiterzuleiten.

¹⁶ Vgl. Vice, R. (2012), S.78

3 Überblick und Kategorisierung von Frameworks am Markt

Im Wortsinn übersetzt heißt Framework so viel wie „Rahmen“ oder „Gerüst“, daher lässt es sich in der Informationstechnologie als sogenanntes „Programmiergerüst“ erklären. Ein Framework selbst ist kein fertiges Programm, sondern stellt seiner Bezeichnung nach nur einen Rahmen dar, in dem eine Anwendung oder ein Programm erstellt werden kann.¹⁷

Frameworks werden vorwiegend in der objektorientierten Programmierung eingesetzt und umfassen Bibliotheken und Komponenten wie Laufzeitumgebungen. Ebenso wird die Designgrundstruktur für die Entwicklung der Bausteine zur Verfügung gestellt. Diese Basisbausteine existieren in Form von abstrakten und konkreten Klassen und unterstützen das Erstellen von Applikationen.¹⁸

Bei der Nutzung eines Frameworks spricht man von Seiten des Programmierers von einer "Umkehrung der Steuerung".¹⁹ Implementierungen werden lediglich registriert, das Framework selbst ist für Nutzung und Steuerung zuständig. Das allgemeine Ziel von Frameworks ist die Entwicklung von Architekturen und Mustern, die wiederverwendet werden können.

3.1 Übersicht von existierenden Frameworks

Es gibt viele verschiedene Arten von Frameworks, die zum Teil nicht strikt getrennt werden, einzelne davon sind:

Data Binding und Datamodel FW: Data Binding ist eine Technik um zwei Datenquellen miteinander zu verbinden und zu synchronisieren. Man muss zwischen einseitigem und zweiseitigem Data Binding unterscheiden. Beim einseitigem Data Binding ist die Datenquelle B mit der Datenquelle A verbunden, A ist jedoch nicht mit B verbunden. Somit haben Veränderungen, die bei A gemacht wurden Einfluss auf B, Veränderungen die bei B gemacht wurden, jedoch keinen Einfluss auf A. Das zweiseitige Data Binding impliziert somit, dass A an B und B an A gebunden ist und somit Veränderungen an einer Datenquelle Einfluss auf die andere Datenquelle haben.²⁰

Eventing Frameworks:²¹ Die einzelnen Komponenten eines Eventing Frameworks haben die Fähigkeit auf bestimmte Aktionen verbundener Komponenten zu reagieren. Dies setzt

¹⁷ Vgl. Wikipedia (2013d)

¹⁸ Vgl. IT Wissen (2013b)

¹⁹ Vgl. Madeyourweb.com (o.J.)

²⁰ Vgl. Friesen, J. (2011)

²¹ Vgl. Kualii (2012)

einen Datenaustausch zwischen den Komponenten voraus. Ein Event kann sowohl vom Server als auch vom Client getriggert werden.

Module Loaders:²² Verschiedene und voneinander unabhängige Module können isoliert behandelt und geladen werden.

Der übergeordnete Nutzen eines Frameworks besteht darin, dass einzelne, vorgefertigte oder parametrisierte Komponenten zu einer wiederverwendbaren Struktur oder Design führen.²³ Ermöglicht wird dies durch die objektorientierte Natur der Framework Komponenten, bei dem Klassen zur Lösung eines bestimmten Problems Inhalt bereitstellen.²⁴

Ein Unterschied innerhalb eines Frameworks wird ersichtlich, wenn es um das Maß an bereits generierten und nicht spezifizierten Komponenten geht. Nicht änderbare Teile werden Frozen-Spots genannt. Der Gegenspieler, der variable Komponenten beinhaltet, wird als Hot-spot bezeichnet.²⁵

Frameworks unterscheiden sich vor allem in ihrem Umfang und in ihrer Funktion als Rahmenwerk. Sie dienen entweder als Gerüst, oder als Baukasten. Die folgende Tabelle zeigt den Unterschied auf:²⁶

Gerüst: Anwendungsframework	Baukasten: Komponentenframework
<ul style="list-style-type: none"> • Um eine funktionierende Anwendung zu entwickeln, müssen die Hot-Spots erst modifiziert werden. • Dies erfordert größere Programmierkenntnisse und stellt einen höheren Aufwand dar. • Begriff: White-Box Framework 	<ul style="list-style-type: none"> • Für die einzelnen Klassen, die benötigt werden, stehen Klassen zur Verfügung. • Hierfür müssen Parameter befüllt werden. • Dies ist die weniger aufwendige Variante, die den Begriff des Black-Box-Framework innehat.

Tabelle 1: Vergleich Anwendungs- und Komponentenframework

²² Vgl. ES WIKI (2013)

²³ Vgl. Hansen, R. Neumann, G. (2002), S. 157

²⁴ Vgl. Ebenda, S. 966

²⁵ Vgl. Frank, U. (2001), S. 204

²⁶ Vgl. Heß, H. (1992), S. 55 ff.

Diese beiden Grundansätze bilden für den Umfang dieser Arbeit die Kategorien, die für Frameworks verwendet wird. Das folgende Kapitel beschäftigt sich mit der genaueren Beschreibung der Ansätze und der Identifizierung zugehöriger Frameworks und Beispielen.

3.1.1 UI Komponentenframeworks

Ein Komponentenframework stellt eine sogenannte „PrebuiltFunctionality“ zur Verfügung, diese instanziiert automatisch bestimmte Komponenten die eine vordefinierte Aufgabe ausführen. Diese Aufgaben sind in den einzelnen Komponenten eingekapselt.²⁷ Komponentenframeworks haben die Eigenschaft einen starken Fokus auf „User Interfaces“ (kurz UI) zu legen. Eine Komponente ist eine Einheit von Software mit den spezifischen Eigenschaften:

- Das Interface ist vordefiniert und spezifiziert, es entstehen Abhängigkeiten
- Sie kann individuell angepasst werden
- Sie kann zusammengesetzt werden
- Sie hat eine hohe Wiederverwendbarkeit: Ohne die Komponenten zu modifizieren, können sie wiederverwendet werden. Ein Beispiel hierfür sind Plug-ins.
- Eine Einheit zur Einführung und Einrichtung ist ein zwingender Bestandteil.

Diese Eigenschaften führen zu einem schnelleren und unkomplizierteren Entwickeln von Anwendungen. Die existierende Anwendung des Frameworks wird durch Komponenten ergänzt und erstellt individuelle Anwendungen über ihre vordefinierte einheitliche Schnittstelle. Die Abbildung stellt schemenhaft den Grundgedanken des UI Komponentenframeworks dar.



Abbildung 4: Beispiel Struktur des UI Komponentenframeworks Twitter Bootstrap²⁸

Um die Erstellung einer kompletten Applikation durch Komponenten zu gewährleisten, ist eine Bibliothek für alle Softwarekomponenten notwendig. Diese stellt Informationen in einer Art Black-Box zur Verfügung, die die Komponente, Regeln für die Integration und Mittel zur

²⁷ Vgl. C2.com (2008)

²⁸ Vgl. TwitterBootstrap (2013)

Verbindung definiert. Das Verbinden der Komponenten wird durch den sogenannten „Glue“ realisiert. Die Black-Box verbirgt hintergründige Informationen und Spezifika vor den Anwendern des Frameworks.²⁹

Um die Hintergründe und die Funktionsweise zu verdeutlichen, dient ein Beispiel eines solchen Frameworks:³⁰

„*Twitter Bootstrap*“ ist ein UI-Komponentenframework mit einer großen Anzahl an wiederverwendbaren Parts, die unter anderem für die Seitennavigation, Alarime oder Menüleisten zuständig sind. Zudem gibt es eine Anleitung über die Schritte, die vorzunehmen sind um eine funktionierende Webanwendung zu erstellen. Informationen über Download, die Dateistruktur und Funktionsumfang sind vorhanden. Zudem werden die verfügbaren Komponenten aufgelistet und der dazugehörige Code wird angezeigt. Die Abbildung zeigt das Beispiel der Generierung von drei Buttons; *<Left>* *<Middle>* *<Right>*. Das Hauptmenü bietet zudem weitere Funktionen und ermöglicht dadurch auch das Verbinden der angewendeten Komponenten durch das sogenannte *Scaffolding*.

Abbildung 5: Beispiel für Komponenten von Twitter Bootstrap³¹

Andere Frameworks, die dieselben UI-Komponenten Eigenschaften besitzen, werden in den nächsten Kapiteln identifiziert.

²⁹ Vgl. Nierstrasz, O./ Lumpe, M. (o.J), S. 3

³⁰ Vgl. TwitterBootstrap (2013)

³¹ Vgl. Ebenda

3.1.2 Anwendungserstellungsframeworks

Als Anwendungsframeworks werden eine Reihe von Funktionalitäten bezeichnet, die für die Anwendungseinstellungen nach dem Beenden eines Programms abgespeichert oder auch als Anwendungsereignisse zur Verfügung gestellt werden.³² In der Computerprogrammierung spricht man von Anwendungsframeworks, wenn Softwareframeworks von Softwareentwicklern zur Implementation von Standardstrukturen einer Anwendung genutzt werden. Anwendungsframeworks wurden durch die ansteigende Benutzung von GUIs bekannt, da diese eine Standardstruktur für Anwendungen benutzen. Es ist wesentlich einfacher GUIs zu entwickeln, wenn ein Standardframework genutzt werden kann, welches die untere Ebene der Codestruktur der Anwendung definiert. Die Programmier Techniken sind größtenteils objektorientiert, sodass die Implementation von Frameworks Teile der schon existierenden Klassen erben kann.³³

Zudem wird ein Anwendungsframework als Softwarebibliothek bezeichnet, die die Struktur zur Unterstützung der Entwicklung von Anwendungen für eine spezielle Umgebung zur Verfügung stellt. Es fungiert als Grundgerüst, d.h. es stellt Struktur und Vorlage bereit, um eine Anwendung zu erstellen und hat die Intention Probleme während des Entwicklungsprozesses der Anwendung zu reduzieren. Diese Reduzierung kann durch die Benutzung von Code, der über unterschiedliche Module der Anwendung genutzt wird, erreicht werden. Anwendungsframeworks werden nicht nur in der Graphical User Interfaces (GUIs) Entwicklung, sondern auch in webbasierten Anwendungen genutzt.³⁴

Ein Beispiel für ein Anwendungsframework sind die Microsoft Foundation Classes (MFC), die in Windowsumgebungen genutzt werden um Anwendungen in C++ zu entwickeln. MFC hat alle Werkzeuge um einen Code automatisch für GUIs zu generieren.

Vorteile von Anwendungsframeworks sind vor Allem die zahlreichen Komponenten, die dem Entwickler eine stückweise Entwicklung erlauben und somit eine Reduktion von Fehlern und Senkung der Entwicklungskosten gewährleisten. Außerdem hilft die Wiederverwendbarkeit von Code und Design im Gebrauch von getesteten Komponenten, die somit die Qualität erhöhen. Die Erweiterbarkeit von anpassbaren Frameworks zur Implementation von Geschäftserfordernissen ist zudem ein Vorteil. Zusätzlich zum Code definieren die vordefinierten Interaktionen zwischen den unterschiedlichen Klassen ein Template, das den Entwicklungsaufwand

³² Vgl. Vb2005 (2005), S. 731

³³ Vgl. Wikipedia (2013)

³⁴ Vgl. Vb2005 (2005), S. 732

reduziert. Dieser Aspekt gewährleistet einen besseren Einstieg in die Softwareentwicklung und hilft der schnellen Anwendungsentwicklung.³⁵

3.2 Überblick über die am Markt befindlichen Frameworks

Die Auswahl an Open Source Frameworks, die im Folgenden dargestellt wird, soll einen Überblick über die am Markt befindlichen Produkte geben. Der Fokus liegt hierbei auf dem Bereich der Desktop Anwendungsentwicklungen, abzugrenzen von der mobilen Variante. Ausgangsbasis für die Übersicht an vorhandenen Frameworks ist der Webauftritt <http://todomvc.com/>. Unter verschiedenen Überbegriffen werden hier Frameworks gelistet. Für das Projekt von Bedeutung sind hier die JavaScript Applikationen, die zum Bau browser-basierter Anwendungen frei zur Verfügung stehen.

Die Plattform, die sich dem Motto verschrieben hat, die Auswahl zwischen den verschiedenen ModelView Frameworks zu erleichtern, vergleicht die Frameworks anhand eines durchgehenden Beispiels. Die unten anstehende Grafik zeigt die Auswahl an Rahmenwerken, die 41 Frameworks, beziehungsweise Kombinationen umfasst.

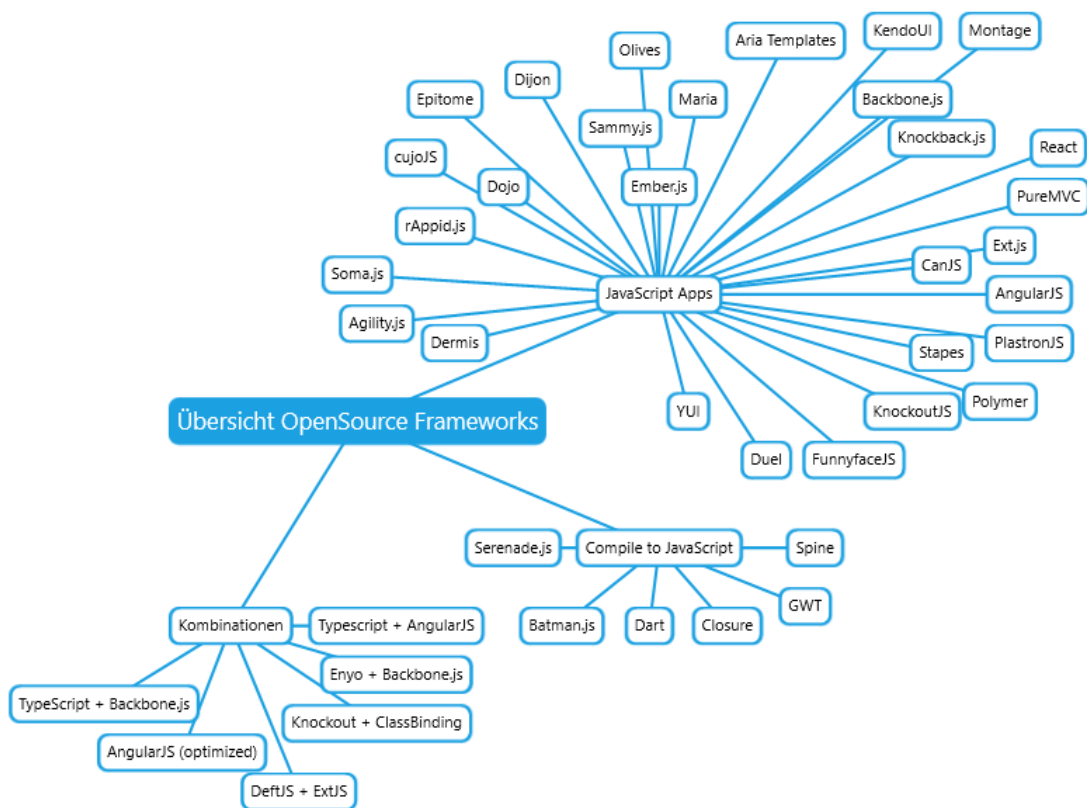


Abbildung 6: Überblick an Open Source Frameworks auf dem Markt³⁶

³⁵ Vgl. Janssen, C. (2010)

³⁶ Vgl. TodoMVC (2013)

Eine Einteilung in die Kategorien, die zuvor analysiert wurden, ist nur vage möglich, da viele Kriterien zweideutig sind und ein separates Bewertungsschema notwendig wäre. Da hierauf kein Fokus der Arbeit liegt, wurde dieser Schritt übersprungen und die Frameworks im nachfolgenden Kapitel eingegrenzt.

Vervollständigt wird das Mindmap mit seiner Auswahl an Anwendungen zusätzlich durch das Framework Twitter Bootstrap. Das Front-end Framework vervollständigt die Liste der Applikationen, die sich der schnelleren und einfacheren Web-Entwicklung gewidmet haben.³⁷

Diese Sammlung von 42 clientseitigen Anwendungen ist zu komplex, um sie im Rahmen dieser Arbeit in einer Marktanalyse zu vergleichen. Daher muss eine Eingrenzung vorgenommen werden, die sich auf den Optionen beruht zu denen durch den Grad an Verbreitung und Anwendung, schon viele Informationen gegeben sind. Dieser Faktor wird die Evaluation anhand des geschaffenen Bewertungsschemas erleichtern und die Funktionsweise dessen repräsentieren.

3.3 Eingrenzung marktrelevanter Frameworks

Da im Rahmen dieser Arbeit die Erstellung eines Bewertungsschemas für Frameworks im Vordergrund steht, soll nur eine kleine Auswahl aus dem Datenpool tatsächlich auf den entwickelten Rahmen angewendet werden. Zur Bestimmung dieser wird das Tool „Google Trends“ verwendet. Die Maxime hinter dieser Vorgehensweise ist, nur die wichtigsten und am meisten verbreiteten Frameworks als Beispielszenarien zu verwenden, da über diese die meisten Informationen erhältlich sind. Als Prämisse gilt folgende Schlussfolgerung:

In der Zeit der World Wide Webs ist Google die am meisten genutzte Suchmaschine mit einem Marktanteil von ca. 90%.³⁸ Durch die Fähigkeit von Google, Häufigkeiten von Suchbegriffen für die letzten Jahre anzuzeigen und in Relation zu setzen, wird die Marktrelevanz der Frameworks in dieser Arbeit von diesen Werten abgeleitet.

Google Trends, welches seine Dienste seit 2004 zur Verfügung stellt, erlaubt den Vergleich und die Darstellung von Suchanfragen verschiedener Kategorien, sowie Gesundheit, Beruf oder Bücher. Die Trend-Suche kann außerdem auf die Faktoren Zeitraum, geografische Aspekte und Web-Inhalte begrenzt werden. Diese Suchkriterien sollen das Stichwort, dessen Trend analysiert wird genauer festlegen, um mögliche Doppeldeutigkeiten der Begriffe zu vermeiden. Dieses Hintergrundwissen ist hier von großer Bedeutung. Das Framework namens *Maria* würde ohne die genauere Sucheinstellung der Kategorie Computer & Elektronik

³⁷ Vgl. Hitz, M. (2013)

³⁸ Vgl. Seo-United (2013)

eine viel höhere Trefferanzahl definieren, da es auch für den Namen Maria stehen kann. Bei zweideutigen Begriffen, die auch in der IT Branche mehrere Bedeutungen mit sich tragen können, wurde zudem in der Suche das Wort *Framework* beigefügt, um die Trefferzahl nicht zu verfälschen.

Aus diesem Grund wurde für die Analyse marktrelevanter Frameworks folgende Grundeinstellung der Häufigkeitsabfrage eingestellt:

Zeitraum	Januar 2010 bis Januar 2013 Dieser Zeitraum gibt einen Einblick über die Trendentwicklung der Frameworks in den letzten 3 Jahren und zeigt zum einen die Beständigkeit und vermeidet somit eine zu kurzfristige Betrachtung.
Geografische Einstellung	Weltweit: Neben den Optionen einzelne Länder für die Analyse zu betrachten, zählt hier die allgemeine und somit weltweite Popularität der Frameworks.
Web-Inhalte	Websuche: Die Häufigkeitsabfrage soll für alle getätigten Websuchen gelten.
Kategorie	Computer und Elektronik

Tabelle 2: Einstellung für die Google Trends Analyse

Google Trends erlaubt den direkten Vergleich von fünf Suchbegriffen, die dann in einem Diagramm auf einer Skala von 0 bis 100 angezeigt werden. 100 steht dabei für das größte Suchinteresse. Die Vorgehensweise für das Projekt bestand darin, die verschiedenen Frameworks durch diese Funktion zu vergleichen. Dabei wurden die jeweils am höchsten bewerteten Frameworks in der Suchliste beibehalten und ein neuer Vergleich vorgenommen. Das resultierende Diagramm aus dem Vergleich kann in der folgenden Abbildung eingesehen werden. Die vollständige Darstellung der Vergleiche befindet sich in den Anhängen 2-7.

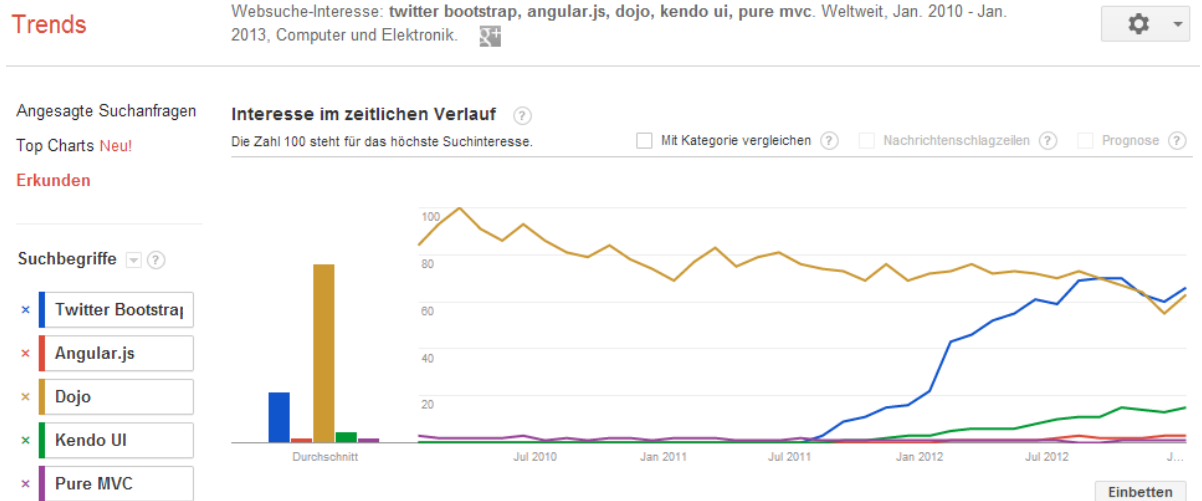


Abbildung 7: Vergleich marktrelevanter Frameworks mit Google Trends

Daraus resultierend lassen sich in absteigender Reihenfolge die höchsten Popularitäten folgender Frameworks feststellen: Twitter Bootstrap, Dojo, Kendo UI, Angular.js und Pure MVC.

4 Erarbeitung eines Kriterienkatalogs zur Bewertung von Frameworks

Das vorige Kapitel zeigt verschiedene Arten von Frameworks auf, die bei der Erstellung einer neuen Anwendung in Betracht gezogen werden können. Nachfolgend muss ein Kriterienkatalog erstellt werden, um Vergleiche zwischen Frameworks transparent und nachvollziehbar zu machen. Dafür wird zunächst eine Vorgehensweise für den Kriterienkatalog mitsamt seiner Bewertung erarbeitet, woraufhin das Bewertungsmodell näher beschrieben wird. Danach werden die eigentlichen Kriterien erarbeitet, ihre Eignung überprüft, Abhängigkeiten geklärt worauf am Ende ein vollständiger Kriterienkatalog mit Bewertungsmodell zur Verfügung steht, den man auf die Beispielframeworks anwenden kann.

4.1 Rechtfertigung und Beschreibung der Vorgehensweise

Eine vergleichende Marktanalyse ist die „systematisch methodische Untersuchung der Stellung einzelner [...] im Marktgeschehen“³⁹ mit Ziel von Schaffung von Transparenz.⁴⁰ Dazu bietet die Wissenschaft vielfältige Methoden. Eine dieser Methoden, die sich in den letzten Jahren bewährt hat und vor allem für den Vergleich von Open Source Frameworks, um die es in dieser Arbeit geht, sinnvoll ist, ist ein Kriterienkatalog. Ziel und Zweck eines Kriterienkatalogs ist die Sammlung und Evaluation einzelner Kriterien, die relevant für die zu entwickelnde Anwendung ist. In dieser speziellen Seminararbeit dient der Kriterienkatalog als Hilfestellung bei der Wahl von desktopbasierten Anwendungsframeworks. Dem Entwickler soll die Entscheidung für ein Framework zur Entwicklung seiner desktopbasierten Anwendungen erleichtert werden und somit ein angenehmerer Einstieg in die Entwicklungsarbeit gewährleistet werden.

Der Kriterienkatalog wurde dieser Seminararbeit als Ziel vorgegeben, seine wissenschaftliche Berechtigung findet er durch seine vielfältige Nutzung in nahezu allen Forschungsgebieten.

Der Aufbau eines Kriterienkatalogs wird von den Autoren dieser Seminararbeit selbst gewählt und begründet. Es werden Kriterien erarbeitet, die entscheidungsrelevant für Open Source Frameworks sind. Diese Kriterien werden im Folgenden evaluiert und konkretisiert, um sie zu rechtfertigen. Daraufhin werden sie gruppiert, um Abhängigkeiten zu finden und Synergien (Zusammenwirken) einzelner Kriterien zu ermitteln. Sie werden an Hand ihrer Relevanz für den Aufbau einer desktopbasierten Anwendung bewertet. Zuletzt wird der Krite-

³⁹ Vgl. Gabler Wirtschaftslexikon (o.J.)

⁴⁰ Vgl. Ebenda

rienkatalog in Form einer Tabelle aufgebaut, wonach der Entwickler lediglich die Kriterien in seinen zu vergleichenden Frameworks mit ihrem Vorhandensein und der Relevanz innerhalb des Kriterienkatalogs abgleichen muss und somit zu seinem Ergebnis kommt. Der Katalog dient somit als „Maske“, die über eine Vielzahl von Frameworks gelegt werden kann und die Entscheidungshilfe in Sachen Effektivität und Effizienz optimiert.

4.2 Beschreibung Bewertungsmodell

Hinter dem Kriterienkatalog muss für eine Auswertung ein Bewertungsmodell liegen. Nicht alle Kriterien haben die gleiche Relevanz und sind gleichermaßen entscheidend für die Entwicklung einer Anwendung. Daher haben sich die Autoren für ein Punktbewertungsverfahren⁴¹, ein sogenanntes Scoring Modell, entschieden. Ein solches Scoring Modell besteht aus sechs verschiedenen Stufen⁴², bestehend aus der Ermittlung der Ziele, der Gewichtung der Ziele, der Vergabe von Punkten für die Varianten, Multiplikation von Gewichten, Ermittlung der gewichteten Punkttotale und einer Sensibilitätsanalyse. Nicht alle genannten Punkte sind unumgänglich für diese Seminararbeit, aber dennoch gilt das Muster als wesentlich.

Aus theoretischer Sicht behandelt der Punkt „Ermittlung der Ziele“ eine „Zielsammlung“⁴³, die möglichst vollständig sein soll. In Hinblick auf die Aufgabe ist dies die Findung der einzelnen Kriterien, die im zweiten Punkte „Gewichtung der Ziele“ gewertet werden und in eine Hierarchie eingeordnet werden. Daraufhin erfolgt die sogenannte „Vergabe von Punkten“, um diese Gewichtung aus Punkt zwei mit Werten zu belegen. Die Autoren haben sich entschieden, den Wert 100% als Gesamtpunktzahl zu verwenden und auf die jeweiligen Gruppierungen herunter zu brechen.

Die nächsten Punkte Multiplikation und Ermittlung der gewichteten Punkttotale beschreibt die praktische Anwendung des Kriterienkatalogs. Mit ausfüllen der jeweiligen Kriterien, die ein Framework hat und der Multiplikation mit der jeweiligen Gewichtung, wird am Ende deutlich, welches Framework am sichersten den Anforderungen des Entwicklers entspricht. Die abschließende Sensibilitätsanalyse dient dem „Variieren von Zielen, Gewichten und Punktwerten und der Überprüfung der Auswirkung auf die bewertete Reihenfolge der Varianten.“⁴⁴

⁴¹ Vgl. Schierenbeck, H. (2008), S. 192

⁴² Vgl. Schmidt, G. (2006), S. 327

⁴³ Vgl. Schierenbeck, H. (2008), S. 193

⁴⁴ Vgl. Ebenda, S. 197

4.3 Erarbeitung der Evaluations-Kriterien

Im Folgenden Kapitel werden die Kriterien für den Kriterienkatalog behandelt und ein abschließendes Ergebnis geliefert. Zuerst werden die Kriterien arbeitet und analysiert, daraufhin erfolgt die Gruppierung anhand der Abhängigkeiten der Kriterien, woraufhin am Ende ein vollständiger Kriterienkatalog zu entnehmen ist.

4.3.1 Findung, Gruppierung und Bewertung der Kriterien

Um die Frameworks in einer zuverlässigen und vollständigen Art und Weise vergleichen zu können müssen Kriterien gefunden werden anhand welcher die einzelnen Frameworks bewertet werden können. Eine Übersicht über alle Kriterien und deren Gewichtung findet sich in Tabelle 3. Folgend sollen nun die einzelnen Kriterien und deren Bedeutung für eine Bewertung näher erläutert werden. Die Kriterien, die in verschiedenen Workshops/Brainstormings und Nachforschung in der Gruppe herausgefunden wurden, sind dafür in die vier Hauptgruppen Generelle Daten, Allgemein, Nutzung und Features aufgeteilt. Die Erläuterung wird anhand der einzelnen Gruppen stattfinden.

Generelle Daten

Die Hauptgruppe „Generelle Daten“ beschreibt die allgemeinen Informationen, die für einen Vergleich von Frameworks notwendig sind. Sie geben sozusagen den ersten Grobüberblick über das Framework und weitere wissenswerte Informationen. Die „Generellen Daten“ sind auch nicht vorgesehen in die Bewertung mit einzufließen, da eine sinnvolle Gewichtung und Bewertung nicht möglich ist.

Informationen die unter der Gruppe „Generelle Daten“ gesammelt werden sind URL, Lizenzmodell, Sprachfamilie und Einsatzbereich. Die URL soll, wie der Name schon sagt, den Link zu der jeweiligen Seite des Frameworks bereitstellen.

Da verschiedenen Frameworks unter unterschiedlichen Lizenztypen stehen ist es unausweichlich, eine Unterscheidung jener Lizenztypen zu machen. Der wichtigste Unterschied hier ist das Open Source Modell, das kostenfrei dem Benutzer zur Verfügung steht und das kommerzielle Framework, bei dem die Nutzung Lizenzgebühren veranschlagt. Diese Unterscheidung wird keinen Einfluss auf diese Arbeit haben, da von vornherein nur Open Source Frameworks behandelt werden, ist aber der Vollständigkeit halber aufgelistet und für eine Weiterverwendung des Kriterienkatalogs nützlich.

Die Sprachfamilie ist eine weitere wichtige Information wenn man versucht Frameworks zu vergleichen, da die Sprachfamilie ein ausschlaggebender Punkt sein kann, besonders auch mit Blick auf die Entwickler, die für die Umsetzung des gewünschten Produktes durch das

Framework verantwortlich sind. So gesehen könnte die Sprachfamilie auch eine Art Kriterium sein, ist aber in diesem Bewertungsverfahren außen vor gelassen, um den Vergleich möglichst auf einer hohen Abstraktionsebene durchzuführen.

Frameworks haben verschiedene Einsatzbereiche. Sei es clientseitig, beim Benutzer, oder serverseitig auf den Servern des Providers des jeweiligen Services oder Produktes. Eine Unterscheidung der Frameworks in diesen beiden Kategorien ist wichtig um sicherzustellen, dass die zu vergleichenden Frameworks auch für den gleichen Einsatzbereich vorgesehen sind. Andererseits wäre ein Vergleich überflüssig und würde auch das Ergebnis der Bewertung verfälschen.

Allgemein

Die „Allgemein“ Gruppe beginnt mit der Beschreibung der eigentlich in die Bewertung einfließenden Kriterien. Sie enthält all die Kriterien, die „um das Framework herum“ ausschlaggebend sind und für einen ausgiebigen Vergleich unausweichlich sind. Darunter fallen die Kriterien: Release Datum, Entwicklungsbeginn, Supportmöglichkeiten, Tutorial, Verbreitung/Reife, Webseitenaktualität und Dokumentation. Eine Kombination aus diesen Kriterien gibt einen guten mehr detaillierten Eindruck des Frameworks. Die Kriterien Gruppe „Allgemein“ fließt mit 30% in die Gesamtwertung ein und haben somit den zweitstärksten Einfluss.

Das Release Datum ist ein wichtiger Anhaltspunkt für die Reife eines Frameworks und hat unausweichlich auch einen Einfluss auf die folgenden Kriterien. Grob kann gesagt werden, je länger ein Framework auf dem Markt ist, also, je länger das Release Datum her ist, umso größer ist die Wahrscheinlichkeit, dass es reifer ist als andere, kürzlich veröffentlichte Frameworks. Andere Einflüsse, wie „aufgetretene und gelöste Fehler“ oder „beantwortete Fragen/Probleme“ sind somit auch wahrscheinlicher schon aufgetreten und können somit entscheidend sein für eine aussagekräftige Bewertung. Das Release Datum kann auch in Verbindung mit dem Kriterium der Webseitenaktualität auf eingestellte Entwicklungen hindeuten. Das Kriterium wird mit 10% in die Bewertung einfließen.

Der Entwicklungsbeginn ist ebenso ein entscheidendes Kriterium für die Reife wie das Release Datum, zeigt allerdings den Reifegrad eines Frameworks noch besser. Spontanentwicklungen besitzen meist einen geringeren Reifegrad als längerfristig entwickelte Produkte: vor 2010: 1.0, 2010-2011: 2.0, 2011 – 2012: 3.0, 2012 – aktuell: 4.0. Der Entwicklungsbeginn fließt, wegen seiner Wichtigkeit, mit 20% in die Bewertung der „Allgemein“ Kriterien ein.

Die Schwierigkeit der Arbeiten mit Frameworks hängt immer davon ab wie gut man bei der Problemlösung unterstützt wird. Seien es bekannte Bugs oder neu aufgetretene Probleme, der Programmierer muss Anlaufstellen haben um diese Probleme zu lösen. Dafür ist es ent-

scheidend ob und wie viele Supportmöglichkeiten es gibt, seien es aktive Foren, Communities oder Mailinglisten. Je mehr desto besser. Die Supportmöglichkeiten fließen mit 10% in die „Allgemein“ Kriterien ein.

Um mit einem neuen Framework umgehen zu können wird immer eine gewisse Arbeitszeit benötigt. Um letzteres zu verringern gibt es sehr häufig Tutorials, die die Einarbeitungszeit merklich verkürzen können und somit die Arbeit mit dem Framework erheblich erleichtern. Die Wahrscheinlichkeit einer schnellen und besseren Einarbeitung steigt. Ob Tutorials vorhanden sind oder nicht fließt mit 10% in die „Allgemein“ Kriterien ein.

Eine aktive Unterstützung und weiterentwickeltes Framework sollte Grundsätzlich gewählt werden wenn man nach einem passenden Framework sucht. Die Webseitenaktualität ist dafür ein guter Indikator. Je aktueller die Webseite umso größer ist die Wahrscheinlichkeit, dass auch das Framework weiterentwickelt und unterstützt wird. Bis zu 3 Monate zurückliegend: 1.0, bis zu 6 Monate zurückliegend: 2.0, bis zu 9 Monate zurückliegend: 3.0, bis zu 12 Monate: 4.0, bis zu 15 Monate: 5.0, darüber hinaus: 6.0

All die bis hierhin genannten Kriterien wirken sich auf die Dokumentation des Frameworks aus, manche mehr, wie zum Beispiel der Entwicklungsbeginn, andere weniger wie zum Beispiel die vorhandenen Tutorials. Alle Kriterien sorgen zusammen für eine Dokumentation über das jeweilige Framework. Je ausführlicher, strukturierter und leichter zu finden die Dokumentation ist, umso besser ist es für den späteren Programmierer und die Bewertung des Frameworks. Daher fließt die Dokumentation mit 20% in die „Allgemein“ Kriterien ein.

Nutzung

Die Nutzung umfasst lediglich drei Kriterien. Diese Kriterien erschienen der Gruppe allerdings so wichtig, dass sie in eine extra Gruppe gefasst wurden um den Einfluss auf die Gesamtbewertung zu erhöhen. Die Kriterien Gruppe „Nutzung“ und die Kriterien „Abhängigkeit“, „Einfachheit der Installation“ und „Bekannte Probleme“, werden mit 20% in die Gesamtbewertung einfließen.

Die Abhängigkeit von Frameworks ist entscheidend für den späteren Einsatzbereich für das Framework und das gesamte Projekt. Wenn ein Framework noch eine Vielzahl anderer Frameworks und Addons benötigt kann der benötigte Aufwand für die Integration rapide steigen. Auch die möglichen Komplikationen, die dann mit den unterschiedlichen Teilen auftreten und beseitigt werden müssen, können den Aufwand und mögliche Kosten, durch den Einsatz von extra Kräften, erhöhen. Je umfangreicher ein einzelnes Framework ist, je weniger es von anderen Komponenten abhängig ist, umso besser wird es in der Bewertung abschneiden.

Die Abhängigkeit der Frameworks geht Hand in Hand mit der Einfachheit der Installation. Je komplexer die Installation umso höher der Zeitaufwand und die daraus resultierenden Kosten. Die Abhängigkeit und die Installation fließen zusammen mit 70% in die Kriterien Gruppe „Nutzen“ ein.

Sind Probleme bekannt mit dem Framework, sollten sie am besten schon gelöst sein, bevor man anfängt mit dem Framework zu arbeiten. Allerdings gibt es Probleme, die immer wieder auftauchen können. Seien es zum Beispiel Kompatibilitätsprobleme oder missverständlich ausgeführte Anleitungen. Wenn es bestehende Probleme gibt für ein Framework, wird es unter diesem Kriterium negativ mit 30% in die Kriterien Gruppe einfließen.

Features

Die Kriterien Gruppe „Features“ schließt all die Kriterien ein, die ausschlaggebend für die Bewertung des Funktionsumfangs eines Frameworks sind. Da dies der ausschlaggebende Punkt für die Auswahl eines Frameworks ist, fließt die Gruppe „Features“ mit 50% in die Gesamtbewertung ein. Die folgenden Kriterien werden näher erläutert: Größe des Frameworks, Browserunterstützung, Anzahl Klassen und Methoden, Unterstützte Datenformate, Widgets, Effekte/Animation, MVC Unterstützung, Nachladen Möglich, Validierbarkeit, Fallbackmöglichkeiten, Data Binding, Workarounds.

Die Größe der Frameworks (Dateigröße) sagt zum einen Teil etwas über den Umfang des Frameworks aus als auch über den Traffic den es erzeugt und den Speicherplatz den es benötigt. Da aber weder Speicher, noch Traffic in der heutigen Zeit kritisch zu sehen sind, liegt das Hauptaugenmerk auf dem größeren Funktionsumfang. Da diese Vermutung nicht immer zutrifft, fließt die Größe des Frameworks lediglich mit 5% in die Gruppe mit ein.

Die Browserunterstützung ist besonders wichtig für ein Framework, wie viel Browser unterstützt es und wie lässt es sich auf andere Browser übertragen. Die Browserunterstützung ist wichtig mit Blick auf die späteren Nutzergruppen, dass man so viele wie möglich mit seiner Applikation unterstützen und erreichen kann. Am besten sollte niemand ausgegrenzt werden. Unter Umständen müssen gewisse Addon-Komponenten hinzugefügt werden um dies zu garantieren. Dieses Kriterium beurteilt wie viele Browser von „Haus aus“ unterstützt werden und fließt mit 10% in die „Features“ Bewertung ein.

Das „Anzahl Klassen und Methoden“ Kriterium bewertet wie viele Klassen und Methoden vorhanden sind und mit dem Framework geliefert werden. Je größer die Anzahl umso höher ist die Wahrscheinlichkeit, dass gewisse Funktionen schneller mit dem Framework implementiert werden können, da auf vorhandenen Bausteine zurückgegriffen werden kann. Das Kriterium wird mit 10% in die „Features“ Bewertung einfließen.

Die Anzahl der unterstützten Datenformate abermals erleichtert die Arbeit mit dem Framework und seinen angrenzenden Komponenten. Entscheidend ist die Frage ob das Framework Text, XML, JSON oder hörsprachige Objekte zum Austausch in Client-Server-Kommunikation unterstützt oder nur die Weitergabe der Rückgabewerte von XMLHttpRequest-API? Das Kriterium wird mit 10% in die Bewertung der Gruppe einfließen.

Um eine anwenderfreundliche Applikation anzubieten ist der Gebrauch von Widgets unausweichlich. Entscheidend ist die Frage ob das Framework dieses schon out-of-the-box mitliefert, wie zum Beispiel Buttons, Autocompleter und andere Fensterwerkzeuge oder dieser Funktionsumfang anders erzielt werden muss. Das Kriterium wird mit 10% in die Bewertung der Gruppe einfließen.

Ebenso interessant für die Bewertung ist zu wissen, ob das Framework gewisse Effekte oder Animationen durch das Framework bereitgestellt werden. Das Kriterium „Effekte/Animationen“ lässt diesen Bestandteil in die Bewertung mit 10% einfließen.

Das Kriterium „MVC Unterstützung“ bewertet inwiefern die MVC-Architektur gefördert wird. Sollte sie gefördert werden, wirkt es sich positiv auf die Bewertung aus, da es die Arbeit des Programmierers übersichtlicher gestaltet, da die unterschiedlichen Schichten, Modell, Präsentation, Steuerung von einander in eigenständige Module unterteilt werden. Dieses Kriterium wird mit 5% in die „Features“ Bewertung einfließen da die Bedeutung in den Augen der Verfasser geringer ausfällt als andere Kriterien.

Manchmal ist es sinnvoll, gewisse Applikationselement später nachzuladen um z.B. den Aufbau der Applikation zu beschleunigen. Dieses „dynamische Nachladen“ fließt mit dem Kriterium „Nachladen möglich“ in die Bewertung mit 5% ein.

Das Kriterium „Validierung“ beschäftigt sich mit der Frage ob das Framework proprietäre Tags unterstützt oder reines (X)HTML verwendet wird. Das Kriterium fließt ebenfalls mit 5% in die Bewertung der Kriterien Gruppe ein.

Da nicht alle Nutzer die aktuellsten Browser benutzen, und da die Applikation möglichst viele Nutzer erreichen sollte, ist es besonders wichtig zu schauen ob eine „Fallback“ Möglichkeit besteht und somit ältere Browser unterstützt werden.

Als letztes Kriterium des Kataloges und für die Auswertung durchaus relevante Funktion, ist die des Data Binding. Hier bietet das Framework die Funktion an, Information und zugehörige Daten aus unterschiedlichen Quellen synchron zuhalten oder nicht. Sollte es diese Funktion anbieten, fließt dies positiv in die Bewertung mit 10% ein. Die Gewichtung liegt bei 10% weil in den Augen der Verfasser dies ein obligatorisches Element in einer Applikation heutzutage darstellt.

4.3.2 Vorstellung des Kriterienkatalogs

Aus dieser detaillierten Vorstellung lässt sich nun ein Kriterienkatalog erschließen, der dem Entwickler die Auswahl seiner Frameworks erleichtern soll. Der Kriterienkatalog wird in einem Excel-Dokument erstellt damit die Bewertungen und Gewichtungen gleich in konkreten Prozentsätzen ausgegeben werden. Durch Multiplikation von der prozentualen Bewertung der vier Gruppen und den einzelnen Kriterien kann eine Gesamtbewertung aller Kriterien des Kataloges erstellt werden (siehe Spalte Bewertung ges.). Durch diese Gliederung ergibt sich ein guter Überblick über alle Kriterien und deren Wichtigkeit.

Gruppierung	Kriterium	Gewichtung	Multiplikation	Bewertung ges.
Generelle Daten			0%	
	URL			
	Lizenzmodell			
	Sprachfamilie			
	Einsatzbereich			
Allgemein			30%	
	Release Datum	10%		3%
	Entwicklungsbeginn	20%		6%
	Supportmöglichkeit	10%		3%
	Tutorien vorhanden?	10%		3%
	Verbreitung/Reife?	20%		6%
	Webseitenaktualität	10%		3%
	Dokumentation vorhanden?	20%		6%
Nutzung			20%	
	Abhängigkeit	30%		6%
	Einfachheit Installation	40%		8%
	Bekannte Probleme	30%		6%
Features			50%	
	Größe des Frameworks	15%		8%
	Browserunterstützung	10%		5%
	Anzahl Klassen und Methoden	10%		5%
	Unterstützte Datenformate	10%		5%
	Widgets	10%		5%
	Effekte/Animation	10%		5%
	MVC Unterstützung	5%		3%
	Nachladen Möglich	5%		3%
	Validierbarkeit	5%		3%
	Fallbackmöglichkeiten	5%		3%
	Data Binding	10%		5%
	Workarounds (Backbutton)	5%		3%
		Summe:	100%	100%

Tabelle 3: Kriterienkatalog mit Gewichtungen

Vergleicht man nun zwei Frameworks miteinander, so kommen weitere Spalten in Betracht, an denen der Entwickler aktiv eintragen muss, ob die Kriterien im zu vergleichenden Framework vorhanden sind oder nicht. Dies geschieht mit einer einfachen 1 und 0 Bewertung. Automatisch kann nach Ausfüllen am Ende abgelesen werden, welches der beiden Frameworks eine höhere Übereinstimmungsrate mit den wichtigsten Kriterien hat. Ab einem Wert von größer als 50 %, ist das Framework ein möglicher Kandidat. Im folgenden fiktiven Beispiel wäre damit Framework 1 stärker in Betracht zu ziehen als Framework 2. Somit dient dieses Excel-File als Grundlage für die Bewertung von Frameworks und kann eine schnelle Unterstützungsleistung bei der Auswahl der Frameworks anhand des Kriterienkatalogs geben.

Framework 1	Punkte	Framework 2	Punkte
vorhanden?		vorhanden?	
1	0,03	1	0,03
0	0	1	0,06
1	0,03	1	0,03
1	0,03	1	0,03
0	0	1	0,06
0	0	0	0
0	0	1	0,06
1	0,06	0	0
0	0	0	0
0	0	1	0,06
1	0,075	0	0
1	0,05	0	0
1	0,05	0	0
1	0,05	0	0
1	0,05	0	0
1	0,05	0	0
1	0,025	1	0,025
1	0,025	1	0,025
1	0,025	1	0,025
1	0,025	1	0,025
0	0	1	0,05
1	0,025	0	0
Summe:	60%	Summe:	48%

Tabelle 4: : Beispielhafter Vergleich zweier Frameworks

5 Anwendung des Kriterienkatalogs

In Kapitel 3.3 basierend auf Google Trends wurde herausgefunden, dass Twitter Bootstrap und Dojo die populärsten Frameworks sind. Daher wird das in Kapitel 4 behandelte Bewertungsmodell und der damit zusammenhängende Kriterienkatalog sowie das Vergleichsmodell im Folgenden auf diese Frameworks angewendet.

Zunächst wird Twitter Bootstrap betrachtet. Die Informationen zu diesem Framework wurden zu einem sehr großen Teil der offiziellen Twitter Bootstrap Webseite <http://twitter.github.io/bootstrap/index.html> entnommen. Alle anderen Informationen sind dementsprechend gekennzeichnet.

Danach wird Dojo Toolkit in Betracht gezogen, dessen Informationen weitestgehend von der offiziellen Dojo Toolkit Webseite <http://dojotoolkit.org> entnommen wurden. Informationen von anderen Quellen sind gekennzeichnet.

Twitter Bootstrap

Grup- pie- rung	Kriterium	Ge- wic- htun- g	Erklärung
Gene- relle Daten			
	URL		http://twitter.github.io/bootstrap/index.html
	Lizenzmo- dell		Open-Source-Projekt
	Sprachfa- milie		Bootstrap enthält auf CSS und HTML basierende Gestaltungs- vorlagen. Zudem sind optional JavaScript- Erweiterungen verfügbar.
	Einsatzbe- reich		Kann in den Bereichen Desktop und mobile Anwendung fin- den. Sowohl als Desktopanwendung als auch auf mobilen Geräten wie Smartphones oder Tablets. ⁴⁵
Allge- ge- mein			
	Release Datum	10%	Seit August 2011 ein Open-Source-Projekt
	Entwick- lungsbe- ginn	20%	Seit August 2011
	Support- möglich- keit	10%	Kein Support auf der Internetseite vorhanden
	Tutorien	10%	Es sind sehr viele unterschiedliche Tutorials vorhanden.

⁴⁵Vgl. stackoverflow (2013)

	vorhanden?		Beispiele: http://www.twitterbootstraptutorial.com http://www.bootstraptutorials.com
	Verbreitung/Reife?	20%	Bootstrap hat sich seit 2011 zum populärsten GitHub-Projekt entwickelt. Aktuelle Version: 2.3.2 (17.Mai 2013)
	Webseitenaktualität	10%	http://twitter.github.io/bootstrap/ sehr aktueller Changelog (17.Mai 2013)
	Dokumentation vorhanden?	20%	Umfassende Dokumentation auf der Twitter Bootstrap Webseite vorhanden mit umfassenden Erklärungen zu jedem Schritt, der gemacht werden kann. http://twitter.github.io/bootstrap/getting-started.html
Nutzung			
	Abhängigkeit	30%	Keine Abhängigkeit von anderen Frameworks
	Einfachheit Installation	40%	Auf der Twitter Bootstrap Webseite direkte Downloadmöglichkeit. Detaillierter Installationsguide auf der Website. http://twitter.github.io/bootstrap/getting-started.html
	Bekannte Probleme	30%	Nicht vorhanden*
Features			
	Größe des Frameworks	15%	500kb
	Browserunterstützung	10%	Unterstützt Mozilla Firefox, Google Chrome, Safari, Opera, Internet Explorer ⁴⁶
	Anzahl Klassen und Methoden	10%	Nicht vorhanden*
	Unterstützte Datenformate	10%	Nicht vorhanden*
	Widgets	10%	Es gibt unterschiedliche Webseiten die Widgets für Bootstrap zu Verfügung stellen: https://bootsnipp.com/index.php/resources http://axutopia.com/twitter-bootstrap/
	Effekte/Animationen	10%	Vorhanden, aber nur kleine Animationen http://twitter.github.io/bootstrap/javascript.html
	MVC Unterstützung	5%	Vorhanden
	Nachladen Möglich	5%	Vorhanden

⁴⁶ Vgl. Github (2011)

	Validierbarkeit	5%	Nicht vorhanden
	Fallbackmöglichkeiten	5%	Vorhanden
	Data Binding	10%	Vorhanden
	Workarounds (Backbutton)	5%	Vorhanden

Tabelle 5: Auswertung nach Kriterienkatalog – Twitter Bootstrap

Dojo

Gruppierung	Kriterium	Gewichtung	Erklärung
Generelle Daten			
	URL		http://dojotoolkit.org
	Lizenzmodell		BSD Lizenz oder Academic Free Lizenz – beide Lizenzen garantieren Rechte Dojo in Open Source und kommerziellen Bereichen zu nutzen und zu modifizieren
	Sprachfamilie		JavaScript
	Einsatzbereich		Dojo findet in den Bereichen Desktop und Mobil Einsatz – als WebApp und als Mobile Framework
Allgemein			
	Release Datum	10%	Seit August 2005
	Entwicklungsbeginn	20%	Seit August 2005
	Supportmöglichkeit	10%	http://dojotoolkit.org/community/ Dojo Community ist eine Art Support, da sich die Mitglieder gegenseitig Fragen beantworten.
	Tutorien vorhanden?	10%	http://dojotoolkit.org/documentation/tutorials/1.9/start/ Hier findet man einige Tutorials und Dokumentationen für Beginner
	Verbreitung/Reife?	20%	Dojo 1.9.0 (13.Mai 2013)
	Webseitenaktualität	10%	Aktuell (Mai 2013 letzte Veränderung)

	Dokumentation vorhanden?	20%	http://dojotoolkit.org/documentation/ Umfangreiche Dokumentation zu allen Themen vorhanden
Nutzung			
	Abhängigkeit	30%	Keine
	Einfachheit Installation	40%	http://dojotoolkit.org/download/ sehr einfache Implementierung – umfangreiche Guides zur Installation vorhanden
	Bekannte Probleme	30%	Nicht vorhanden*
Features			
	Größe des Frameworks	15%	1,38MB
	Browserunterstützung	10%	Firefox 3.6-8 (Dojo 1.7.2+ also supports Firefox 9-10), Safari 5.0-5.1, Chrome 13-15, IE 6-9, Opera 10.50-11.50 http://dojotoolkit.org/reference-guide/1.9/releasenotes/1.7.html
	Anzahl Klassen und Methoden	10%	Nicht vorhanden*
	Unterstützte Datenformate	10%	Nicht vorhanden*
	Widgets	10%	Auf der offiziellen Dojo Seite sind umfangreiche Widgets zu Verfügung gestellt: http://dojotoolkit.org/features/desktop
	Effekte/Animationen	10%	Vorhanden http://dojotoolkit.org/documentation/tutorials/1.6/effects/
	MVC Unterstützung	5%	Vorhanden
	Nachladen Möglich	5%	Vorhanden
	Validierbarkeit	5%	Vorhanden
	Fallbackmöglichkeiten	5%	Vorhanden
	Data Binding	10%	Vorhanden
	Workarounds (Backbutton)	5%	Vorhanden

* nach aufwendiger Suche keine Informationen vorhanden

Tabelle 6: Anwendung des Kriterienkatalogs - Dojo

Auf die Anwendung des Kriterienkatalogs auf die Frameworks Twitter Bootstrap und Dojo folgt der direkte Vergleich beider Frameworks.

Kriterium	Twitter Bootstrap	Punkte	Dojo Toolkit	Punkte
	vorhanden?		vorhanden?	
URL				
Lizenzmodell				
Sprachfamilie				
Einsatzbereich				
Release Datum	1	0.03	1	0.03
Entwicklungsbeginn	1	0.06	1	0.06
Supportmöglichkeit	0	0	1	0.03
Tutorial vorhanden?	1	0.03	1	0.03
Verbreitung/Reife?	1	0.06	1	0.06
Webseitenaktualität	1	0.03	0	0
Dokumentation vorhanden?	1	0.06	1	0.06
Abhängigkeit	0	0	0	0
Einfachheit Installation	1	0.08	1	0.08
Bekannte Probleme	0	0	0	0
Größe des Frameworks	1	0.075	1	0.075
Browserunterstützung	1	0.05	1	0.05
Anzahl Klassen und Methoden	0	0	0	0
Unterstützte Datenformate	0	0	0	0
Widgets	1	0.05	1	0.05
Effekte/Animation	1	0.05	1	0.05
MVC Unterstützung	1	0.025	1	0.025
Nachladen Möglich	1	0.025	1	0.025
Validierbarkeit	0	0	1	0.025
Fallbackmöglichkeiten	1	0.025	1	0.025
Data Binding	1	0.05	1	0.05
Workarounds (Backbutton usw)	1	0.025	1	0.025
Summe:		73%	Summe:	75%

Tabelle 7: Gegenüberstellung Twitter Bootstrap und Dojo

Anhand der obigen Tabelle wird ersichtlich, dass beide Frameworks nahezu die gleiche Bewertung im Vergleich erhalten. Twitter Bootstrap erreicht eine Prozentzahl von 73% und Dojo eine Prozentzahl von 75%. Nach dem direkten Vergleich sollte man Dojo bevorzugen, jedoch ist der Unterschied so gering, dass man beide Frameworks in Betracht ziehen kann, da beide in dem festgelegten Rahmen von 50% und aufwärts liegen. In diesem Fall kommt es auf die Natur des Anwenders bezüglich des Programmieraufwands, der für die Erstellung der

Web Applikation notwendig ist, an. Wie in den vorangehenden Kapiteln beschrieben, sind die Programmierkenntnisse des Anwenders beim Anwendungsframework mehr gefordert und erlauben zusätzlich größere Spielräume im Vergleich zum Komponentenframework. Somit sollte der Anwender in diesem Fall entscheiden, welches Framework für seinen Einsatzbereich in Bezug auf die Komponenten am besten geeignet ist.

6 Fazit

Ziel dieser Seminararbeit war es, die Basis für eine Marktanalyse für den Vergleich clientseitiger Open Source Frameworks zum Bau von browserbasierten Desktop-Anwendungen zu erstellen. Dazu wurde ein Kriterienkatalog erarbeitet, der als Bewertungsgrundlage für die am Markt befindlichen Anwendungsframeworks dienen soll und somit als große Hilfe für die Marktstudie gilt.

Als Vorarbeit für diesen Kriterienkatalog lag es an den Autoren, sich mit den theoretischen Aspekten von browserbasierten Desktop-Anwendungen auseinanderzusetzen, den grundlegenden Technologien und Architekturmodellen. Sodann wurde sich das Thema Frameworks näher betrachtet. Am Markt befindliche Frameworks wurden in Komponentenframeworks, Anwendungserstellungsframeworks und Teilaufgabenframeworks kategorisiert. Anschließend wurde ein Kriterienkatalog für die Bewertung von Frameworks erarbeitet. Dieser beinhaltet die Erörterung von Kriterien und deren Gewichtung anhand ihrer Wichtigkeit für die Entwicklung von Anwendungen. In einem abschließenden exemplarischen Vergleich werden zwei Frameworks anhand der Kriterien im Katalog gegenübergestellt und bewertet.

In Anbetracht der Zeitspanne von 24 Vorlesungs-Stunden, in denen das Projekt gestemmt werden sollte, waren weiterführende Erweiterungen des Kriterienkatalogs nicht möglich. Dennoch erlaubt es diese Version, das Ziel, eine Marktstudie von Open Source Frameworks, zu verwirklichen. Die Vielfältigkeit der vielzähligen Frameworks stellte sich ebenso als Herausforderung dar, wie auch die Erörterung von unterschiedlichsten Bewertungskriterien, um eine allgemeingültige Bewertungsaussage treffen zu können. Das vorliegende Ergebnis trifft das gewünschte Resultat der Aufgabenstellung. Mit diesem Kriterienkatalog ist eine Marktanalyse für Open Source Frameworks möglich.

Ausblickend lässt sich weiter sagen, dass der Kriterienkatalog immer wieder sowohl vertiefen, als auch erweitern lässt und weiter nach den verschiedenen Typen der Frameworks aufgeteilt werden kann, was in dieser Arbeit aufgrund der knappen Zeit nicht zu realisieren war.

Aufgrund der wachsenden Dominanz, Dienstleistungen aus dem Internet zu beziehen oder bereitzustellen, gilt die Arbeit mit Open Source Frameworks als zukunftsweisend und angebracht.

Anhang

Anhangverzeichnis

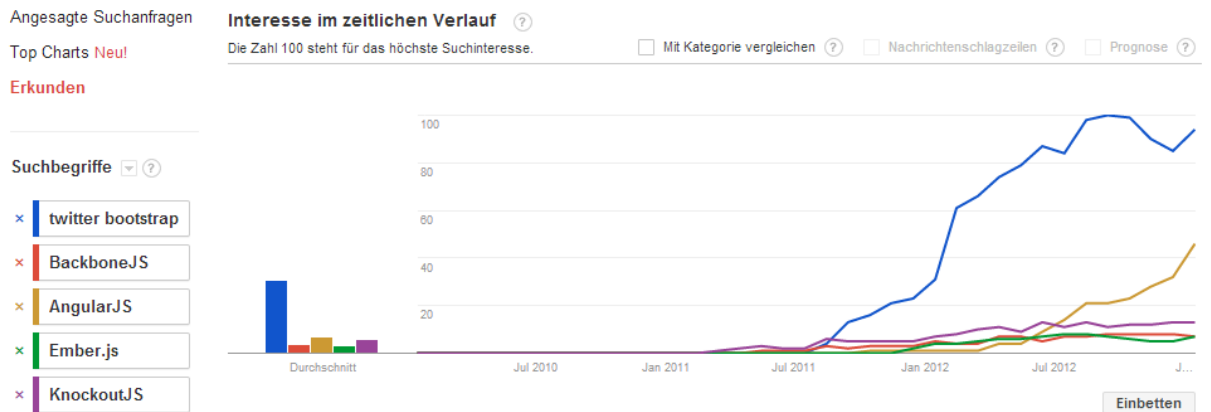
Anhang 1: Auswertung des Kriterienkatalogs	35
Anhang 2: Trendanalyse Frameworks Teil I	36
Anhang 3: Trendanalyse Frameworks Teil II	37
Anhang 4: Trendanalyse Frameworks Teil III	38
Anhang 5: Trendanalyse Frameworks Teil IV	39
Anhang 6: Trendanalyse Frameworks Teil V	40
Anhang 7: Trendanalyse Frameworks Teil VI	41

Anhang 1: Auswertung des Kriterienkatalogs

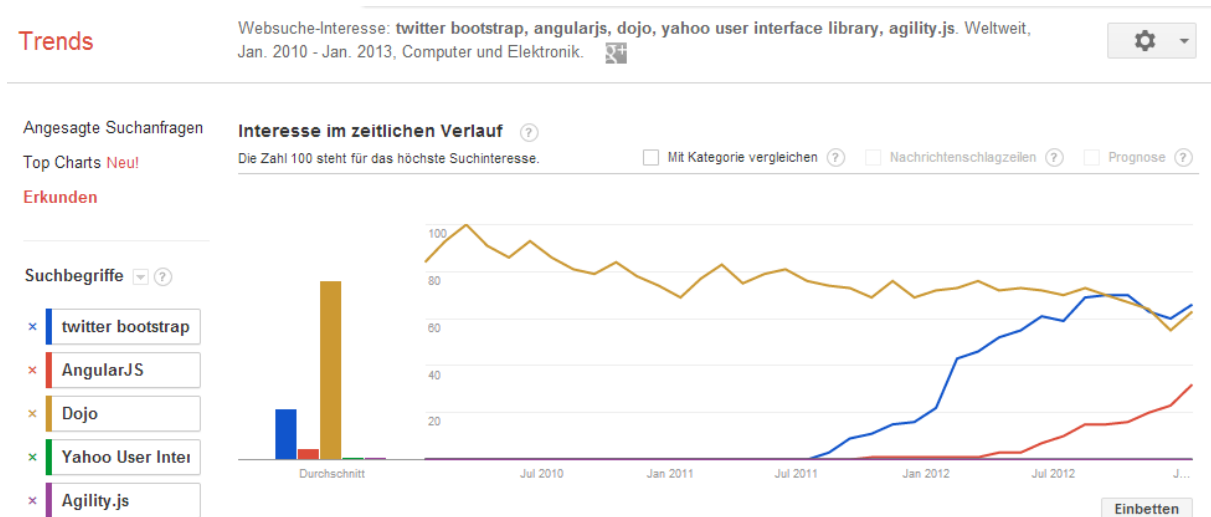
Kriterienkatalog													
Gruppierung	Kriterium	Gewichtung	Multiplikation	Summe gesamt	Framework 1	Punkte	Framework 2	Punkte					
Generelle Daten			0%		vorhanden?		vorhanden?						
	URL												
	Lizenzmodell												
	Sprachfamilie												
	Einsatzbereich												
Allgemein			30%										
	Release Datum	10%		3%	1	0,03	1	0,03					
	Entwicklungsbeginn	20%		6%	0	0	1	0,06					
	Supportmöglichkeit	10%		3%	1	0,03	1	0,03					
	Tutorial vorhanden?	10%		3%	1	0,03	1	0,03					
	Verbreitung/Reife?	20%		6%	0	0	1	0,06					
	Webseitenaktualität	10%		3%	0	0	0	0					
	Dokumentation vorhanden?	20%		6%	0	0	1	0,06					
Nutzung			20%										
	Abhängigkeit	30%		6%	1	0,06	0	0					
	Einfachheit Installation	40%		8%	0	0	0	0					
	Bekannte Probleme	30%		6%	0	0	1	0,06					
Features			50%										
	Größe des Frameworks	15%		8%	1	0,08	0	0					
	Browserunterstützung	10%		5%	1	0,05	0	0					
	Anzahl Klassen und Methoden	10%		5%	1	0,05	0	0					
	Unterstützte Datenformate	10%		5%	1	0,05	0	0					
	Widgets	10%		5%	1	0,05	0	0					
	Effekte/Animation	10%		5%	1	0,05	0	0					
	MVC Unterstützung	5%		3%	1	0,03	1	0,03					
	Nachladen Möglich	5%		3%	1	0,03	1	0,03					
	Validierbarkeit	5%		3%	1	0,03	1	0,03					
	Fallbackmöglichkeiten	5%		3%	1	0,03	1	0,03					
	Data Binding	10%		5%	0	0	1	0,05					
	Workarounds (Backbutton usw)	5%		3%	1	0,03	0	0					
		Summe:	100%	100%	Summe:	60%	Summe:	48%					

Anhang 2: Trendanalyse Frameworks Teil I

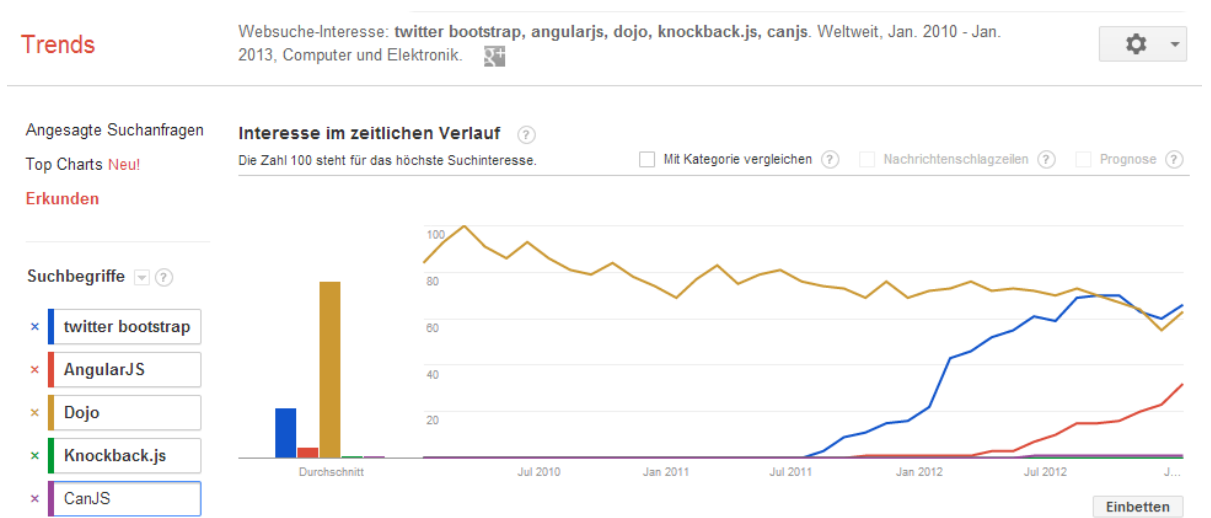
Vergleich: Twitter Bootstrap, BackboneJS, AngularJS, Ember.js, KnockoutJS



Vergleich: Twitter Bootstrap, AngularJS, Dojo, Yahoo User Interface (YUI), Agility.js

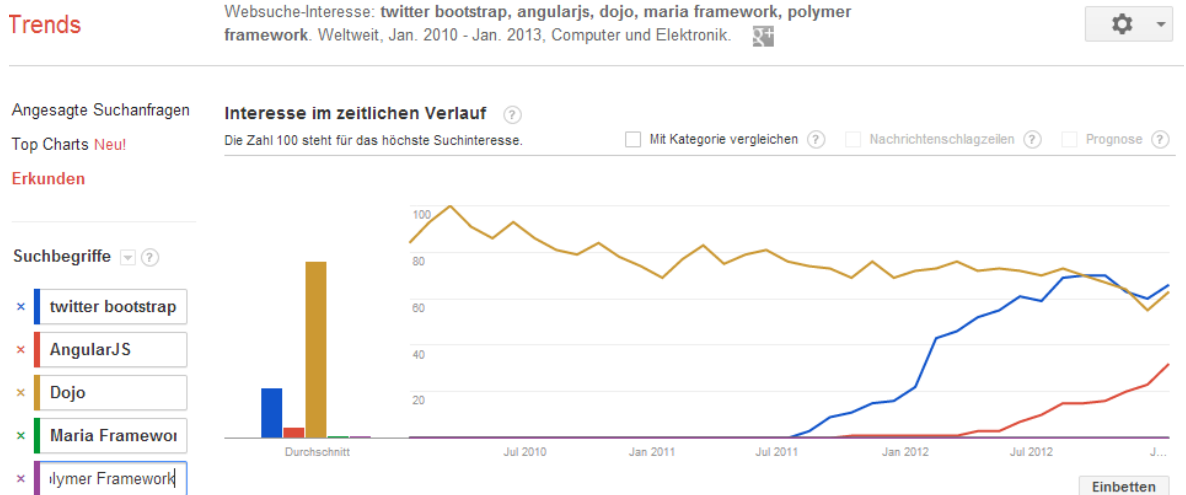


Vergleich: Twitter Bootstrap, AngularJS, Dojo, Knockback.js, CanJS

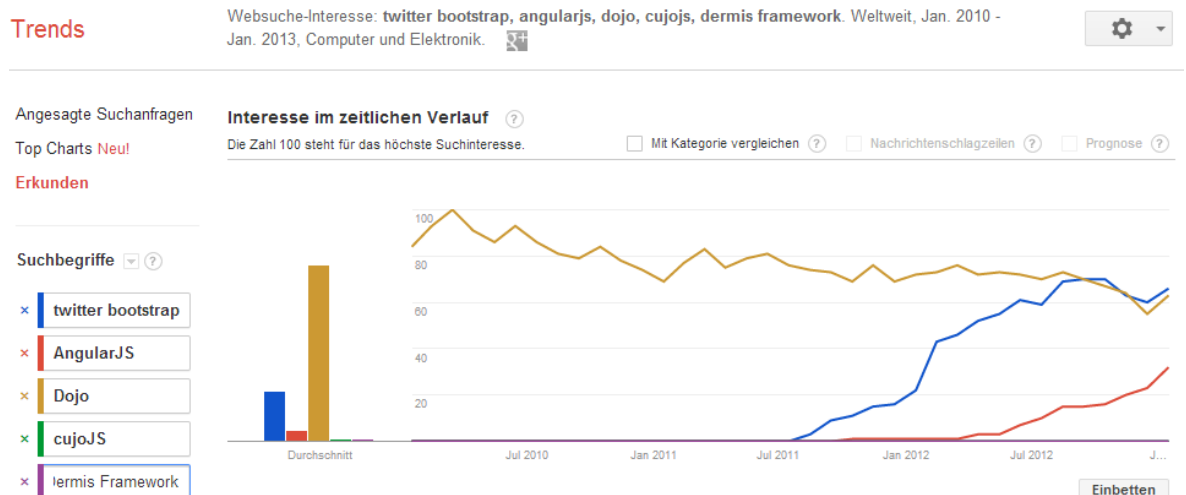


Anhang 3: Trendanalyse Frameworks Teil II

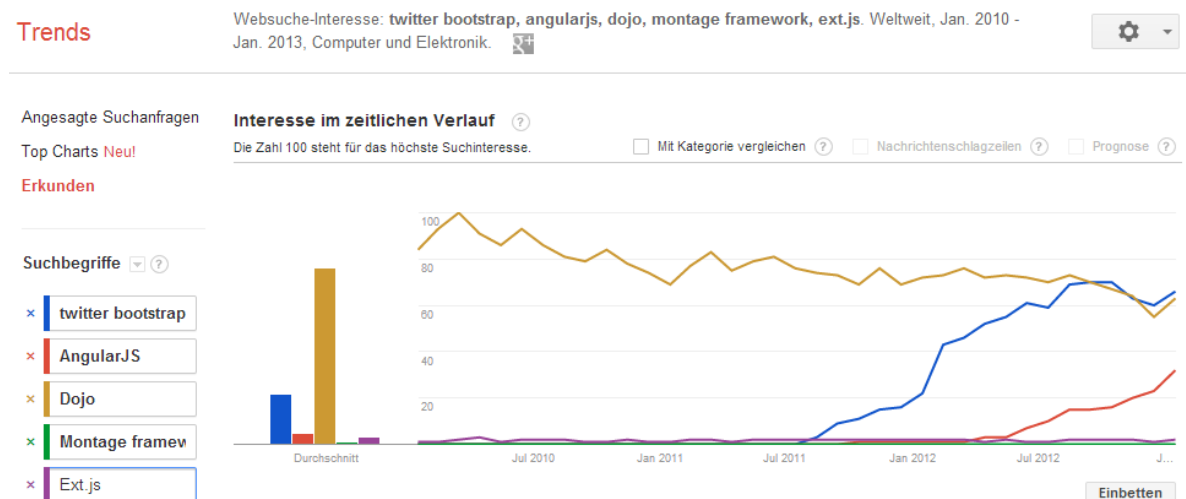
Vergleich: Twitter Bootstrap, AngularJS, Dojo, Maria Framework, Polymer Framework



Vergleich: Twitter Bootstrap, AngularJS, Dojo, cujoJS, Dermis Framework

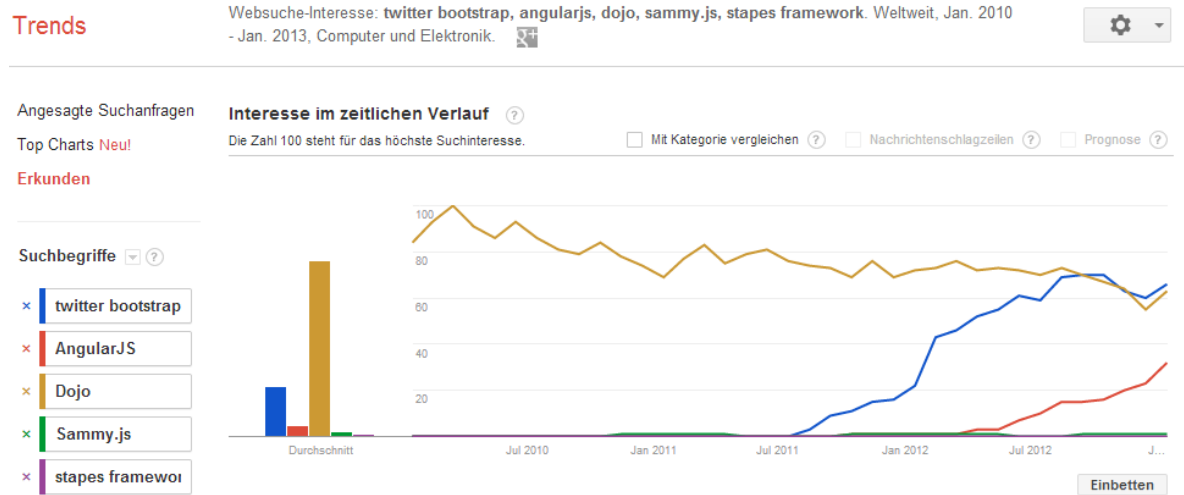


Vergleich: Twitter Bootstrap, AngularJS, Dojo, Montage Framework, Ext.js

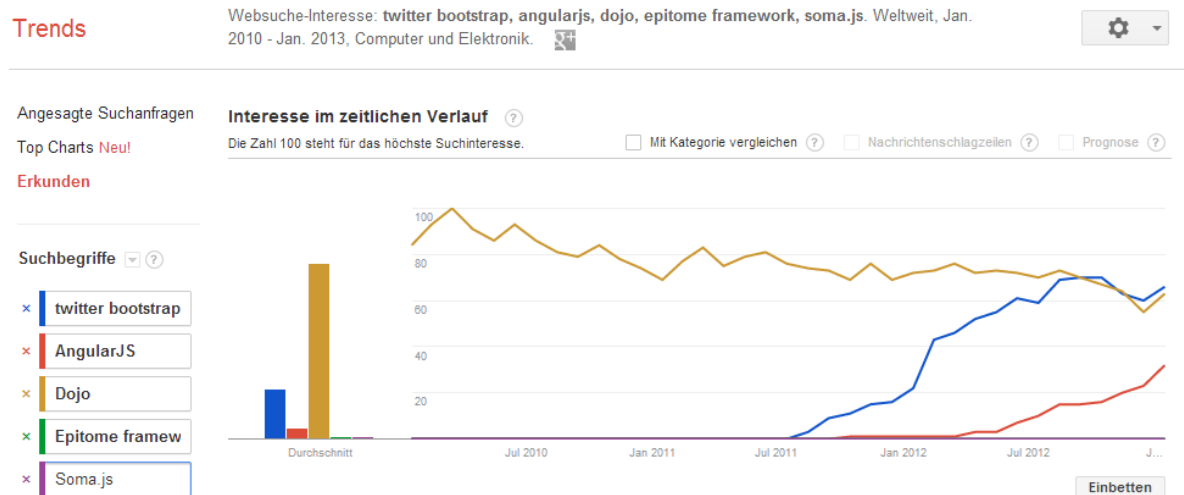


Anhang 4: Trendanalyse Frameworks Teil III

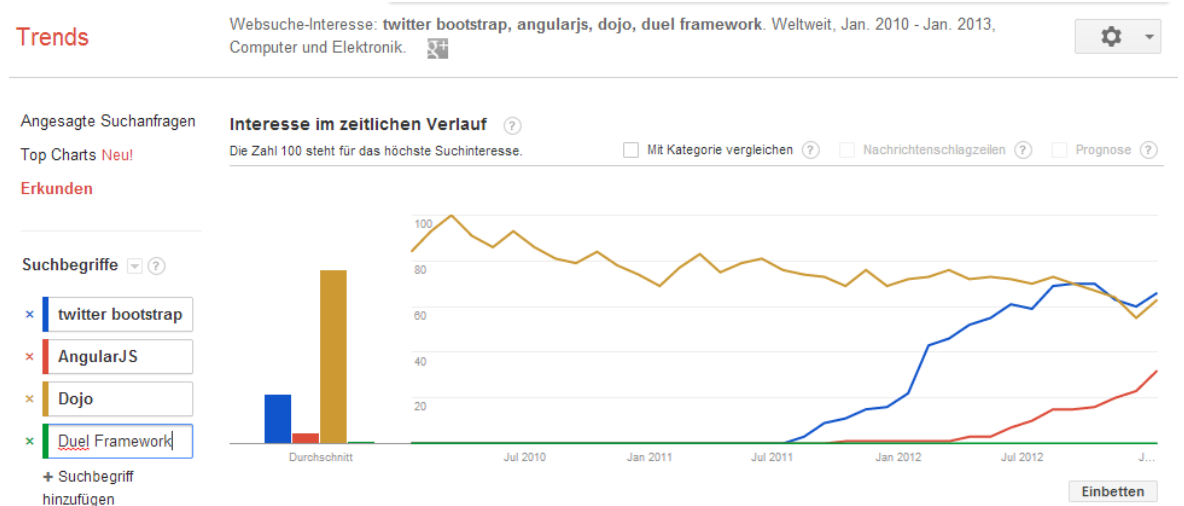
Vergleich: Twitter Bootstrap, AngularJS, Dojo, Sammy.js, stapes framework



Vergleich: Twitter Bootstrap, AngularJS, Dojo, Epitome Framework, Soma.js

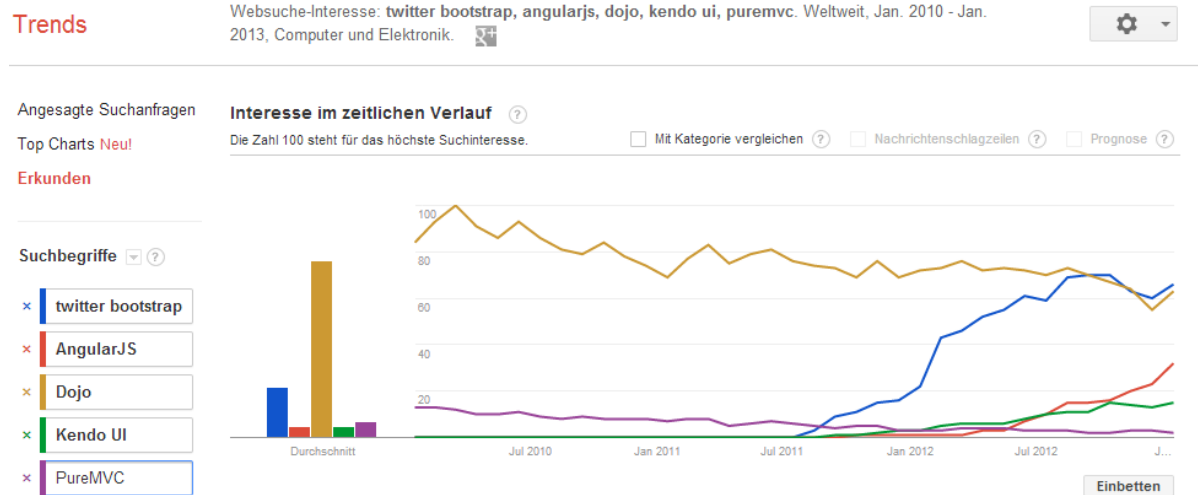


Vergleich: Twitter Bootstrap, AngularJS, Dojo, Duel Framework

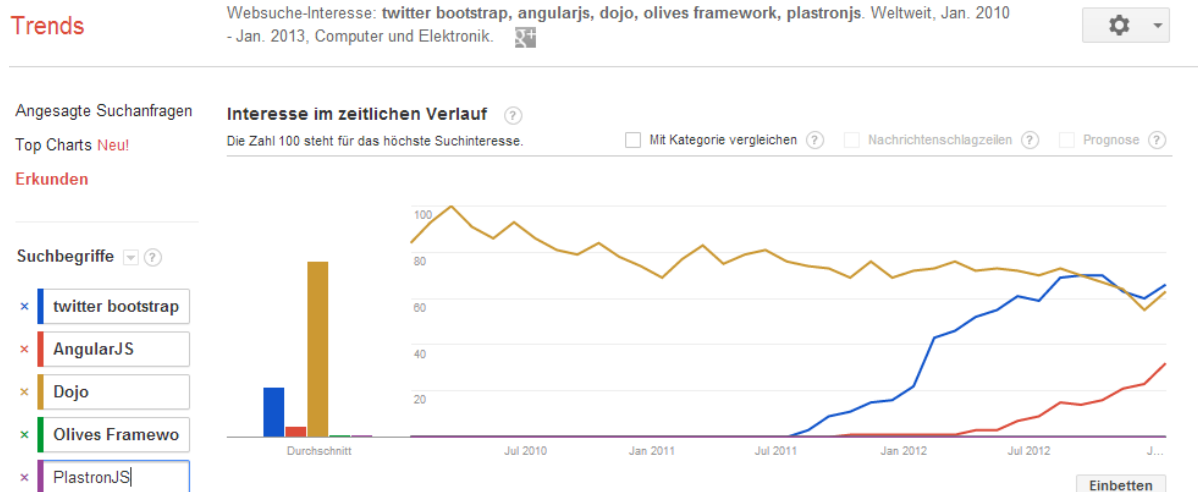


Anhang 5: Trendanalyse Frameworks Teil IV

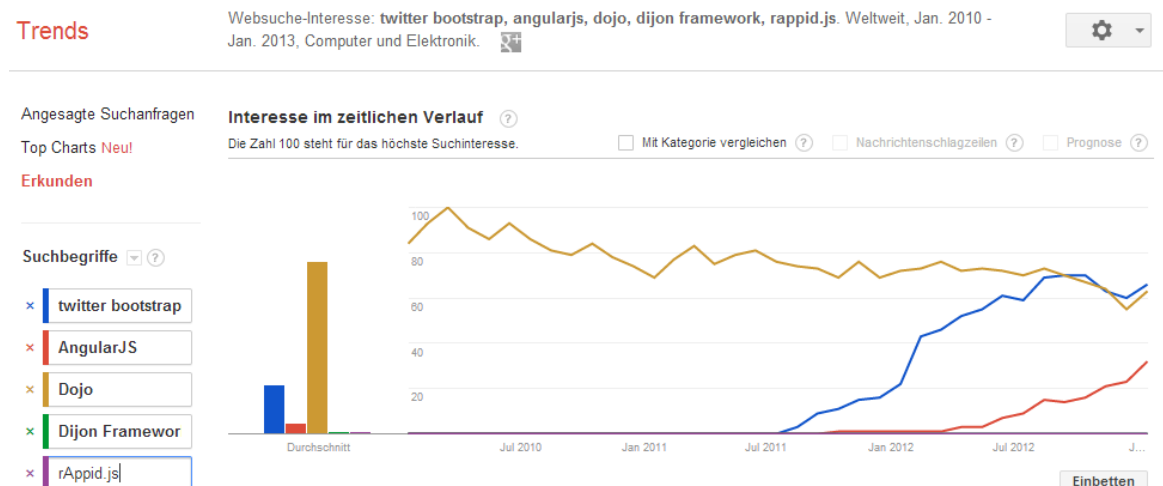
Vergleich: Twitter Bootstrap, AngularJS, Dojo, Kendo UI, Pure MVC



Vergleich: Twitter Bootstrap, AngularJS, Dojo, Olives Framework, PlastronJS

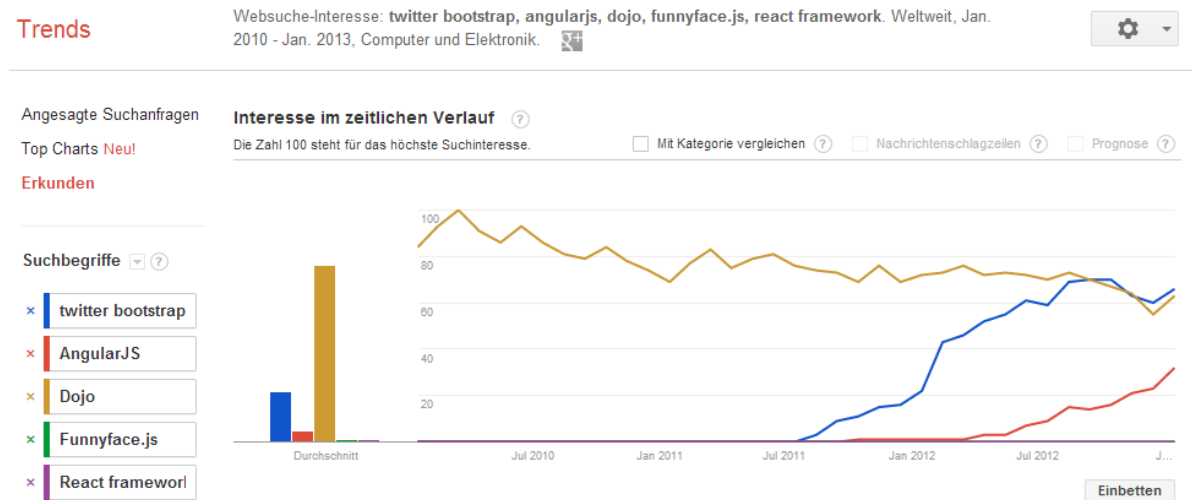


Vergleich: Twitter Bootstrap, AngularJS, Dojo, Dijon Framework, rAppid.js

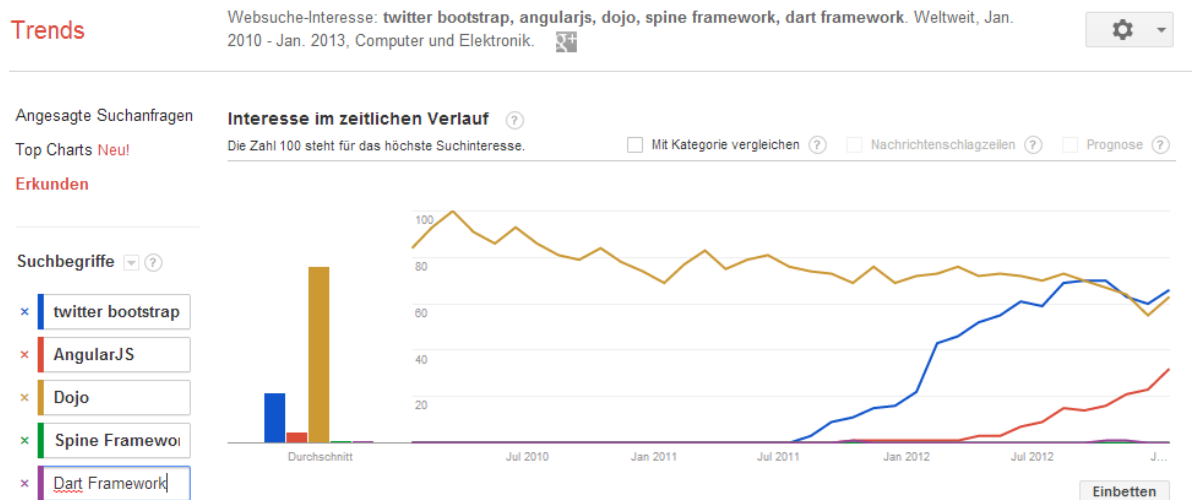


Anhang 6: Trendanalyse Frameworks Teil V

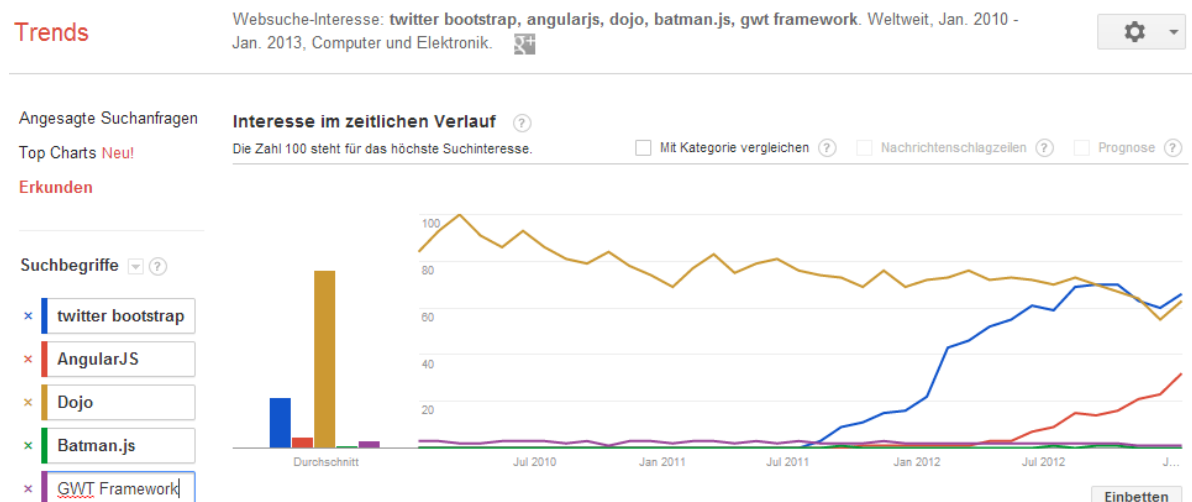
Vergleich: Twitter Bootstrap, AngularJS, Dojo, Funnyface.js, React Framework



Vergleich: Twitter Bootstrap, AngularJS, Dojo, Spine Framework, Dart Framework

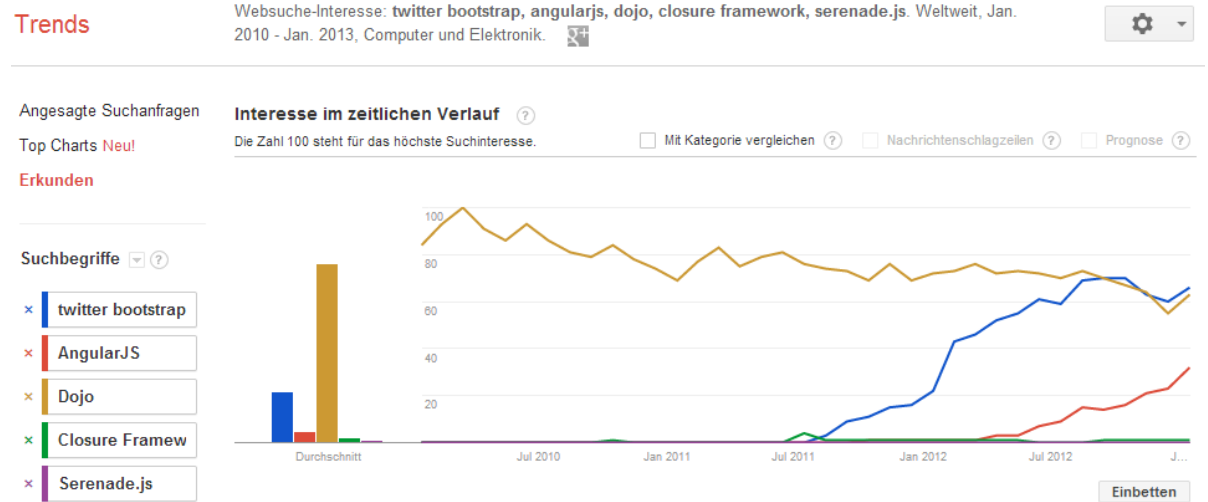


Vergleich: Twitter Bootstrap, AngularJS, Dojo, Batman.js, GWT Framework



Anhang 7: Trendanalyse Frameworks Teil VI

Vergleich: Twitter Bootstrap, AngularJS, Dojo, Closure Framework, Serenade.js



Quellenverzeichnisse

Literaturverzeichnis

- Adams, K. / Agesen, O. (2006): A Comparison of Software and Hardware Techniques for x86 Virtualization, in: Proceedings of the 12th international conference on Architectural support for programming languages and operating systems, Oktober 2006
- Frank, U. (2001): Framework, in: Mertens, P. et al (Hrsg.): Lexikon der Wirtschaftsinformatik, 4. Auflage, Springer Verlag: Berlin
- Hansen, R./ Neumann, G. (2002): Wirtschaftsinformatik I – Grundlagen betrieblicher Informationsverarbeitung, 8. Auflage, Lucius & Lucius: Stuttgart
- Henning, P. / Vogelsang, H. (2007): Programmiersprachen, 2. Auflage, Carl Hanser Verlag: München
- Heß, H. (1992): Wiederverwendung von Software: Framework für betriebliche Informationssysteme, Gabler: Wiesbaden
- Langner, T./Reiberg, D. (2006): J2EE und JBoss: Grundlagen und Profiwissen, Carl Hanser Verlag: München
- Lawson, B./ Shar, R. (2011): HTML5: Eine Einführung für Umsteiger, Addison-Wesley: München
- Lott, J / Patterson, D. (2007): ActionScript 3 with Design Patterns, Peachpit Press: Berkeley
- Schierenbeck, H./ Wöhle, C. (2008): Grundzüge der Betriebswirtschaftslehre, Oldenbourg Wissenschaftsverlag GmbH München, 17. Auflage
- Schmidt, G. (2006): Organisatorische Grundbegriffe, ibo Schriftenreihe, Band 3, 13. Auflage, Gießen
- Steyer, R. (2010): Das JavaScript-Handbuch: Einführung, Praxis und Referenz, Addison-Wesley: München
- Vice, R. (2012): MVVM Survival Guide for Enterprise Architectures in Silverlight and WPF, Packt Publishing Ltd.:Birmingham

Verzeichnis der Internet- und Intranet-Quellen

- BpB (2013): Bundeszentrale für politische Bildung, Dossier 2013
<http://www.bpb.de/gesellschaft/medien/opensource/>
 Abruf: 25.06.2013
- C2.com (2008): Component Frameworks,
<http://c2.com/cgi/wiki?ComponentFramework>
 Abruf: 22.06.2013
- ES WIKI (2013): Module Loaders, loaders
http://wiki.ecmascript.org/doku.php?id=harmony:module_loaders
 Abruf: 27.06.2013
- Friesen, J. (2011): Java World: Data Binding,
<http://www.javaworld.com/community/node/7580>
 Abruf: 20.05.2013
- Gabler Wirtschaftslexikon (o.J.): Stichwort Marktanalyse
<http://wirtschaftslexikon.gabler.de/Archiv/378/marktanalyse-v7.html>, Abruf: 27.06.2013
- GitHub (2011) Browser Compatibility
<https://github.com/twitter/bootstrap/wiki/Browser-Compatibility>
 Abruf: 27.06.2013
- HS-Augsburg (2011): Kommunikation zwischen MVC-Modulen
<http://glossar.hs-augsburg.de/Model-View-Controller-Pattern>
 Abruf: 12.05.2013
- IT Wissen (2013a): AJAX (asynchronous JavaScript and XML),
<http://www.itwissen.info/definition/lexikon/asynchronous-JavaScript-and-XML-AJAX.html>
 Abruf: 20.06.2013
- IT Wissen (2013b): Das große Online-Lexikon für Informationstechnologie,
 Stichwort: Framework,
<http://www.itwissen.info/definition/lexikon/Framework-framework.html>
 Abruf: 10.06.2013

- Kuali (2012): UIF Eventing Framework,
<https://wiki.kuali.org/display/KULRICE/UIF+Eventing+Framework>
Abruf: 20.06.2013
- Madeyourweb.com (o.J.): Was sind Frameworks,
<http://www.madeyourweb.com/webentwicklung/was-sind-frameworks.html>
Abruf: 11.06.2013
- Media Engineering (2013): WebApps – Browser-basierte Anwendungen,
<http://www.11media.net/webapps-browser-basierte-anwendungen/>
Abruf: 10.05.2013
- Nierstrasz, O./ Lumpe, M. (o.J.): Komponenten, Komponentenframeworks und Gluing
<http://scg.unibe.ch/archive/papers/Nier97aKomponentenUndGluing.pdf>
Abruf: 19.06.2013
- Seo-United (2013): Suchmaschinenoptimierung, Beratung & Consulting,
<http://www.seo-united.de/suchmaschinen.html>
Abruf: 15.06.2013
- Stackoverflow (2013): Same style for desktop and tablet with Twitter Bootstrap,
<http://stackoverflow.com/questions/15452715/same-style-for-desktop-and-tablet-with-twitter-bootstrap>
Abruf: 27.06.2013
- Techopedia (2010): Application Framework,
<http://www.techopedia.com/definition/6005/application-framework>
Abruf: 23.05.2013
- TodoMVC (2013): Helping you select an MV framework, <http://todomvc.com/>
Abruf: 10.05.2013
- TwitterBootstrap (2013): Bootstrap, <http://twitter.github.io/bootstrap/>
Abruf: 19.06.2013
- VB2005 (2010): Das Anwendungsframework,
<http://vb2005.de/VB2005/F26%20-%20Anwendungsframework.pdf>
Abruf: 20.05.2013

- Wikipedia (2013a): Stichwort: Webanwendungen,
<http://de.wikipedia.org/wiki/Webanwendung>
Abruf 10.05.2013
- Wikipedia (2013b): Stichwort: MVVM
http://de.wikipedia.org/wiki/Model_View_ViewModel
Abruf: 12.05.2013
- Wikipedia (2013c): Stichwort: Application Framework
http://en.wikipedia.org/wiki/Application_framework
Abruf: 12.05.2013
- Wikipedia (2013d): Stichwort: Framework
<http://de.wikipedia.org/wiki/Framework>
Abruf: 10.05.2013

Gesprächsverzeichnis

- Hitz, M. (2013): Einführungsgespräch und Dossier 10.05.2013, DHBW
Stuttgart

Literatur-Recherche: Web Engineering

Modellgestützte Generierung von Oberflächen

Seminararbeit

vorgelegt am 02.7.2013

Fakultät Wirtschaft

Studiengang Wirtschaftsinformatik

Kurs WIWI2010I

von Janina Höchner, Robert Kromholz, Sarah Kieninger, Marco Berger

Inhaltsverzeichnis

Inhaltsverzeichnis	II
Abkürzungsverzeichnis	IV
Abbildungsverzeichnis	V
Tabellenverzeichnis	VI
Glossar	VII
1. Einführung	- 1 -
1.1. Problemstellung	- 1 -
1.2. Ziele der Arbeit	- 2 -
1.3. Methodik der Arbeit.....	- 2 -
2. Einführung in das modellgestützte Web Engineering	- 3 -
2.1. Was bedeutet MDA/ MDD.....	- 5 -
2.2. Multikanalarchitekturen	- 6 -
3. Grundlagen und Besonderheiten der Modellierung von Web-Anwendungen.....	- 7 -
4. Vorstellung der verschiedenen Methoden	- 9 -
4.1. Übersicht der verschiedenen Ansätze.....	- 9 -
4.2. Vorstellung ausgewählter Ansätze	- 12 -
4.2.1. UWE	- 12 -
4.2.2. WebML	- 14 -
4.2.3. OOHDM.....	- 17 -
4.2.4. OOWS	- 21 -
4.3. Literatur zu anderen Ansätzen	- 22 -
5. Ansätze zur Generierung von Weboberflächen.....	- 26 -
5.1. Weboberflächen.....	- 26 -
5.2. UWE	- 27 -
5.3. WebML:	- 27 -
5.4. OOHDM.....	- 28 -
5.5. OOWS	- 29 -
5.6. XUL und XFORMS.....	- 31 -

6. Zusammenfassung und Literaturübersicht	- 35 -
7. Fazit.....	- 47 -
Literaturverzeichnis	VIII
Internetverzeichnis	X

Abkürzungsverzeichnis

OOHDM

Object Oriented Hypertext

UWE

UML-based Web Engineering

WebML

Web Modeling Language

MDWE

Model Driven Web Engineering

OOWS

Object Oriented Web Solution

Abbildungsverzeichnis

Abbildung 1: Das Schema der MDWE Methoden	- 5 -
Abbildung 2: Anforderungen an die Modellierung von Web-Anwendungen.....	- 7 -
Abbildung 3: Übersicht über Modellierungsmethoden für Web-Anwendungen	- 9 -
Abbildung 4: Übersicht und Kategorisierung der Ansätze	- 11 -
Abbildung 5: Zusammenhänge zwischen WebML Modellen	- 16 -
Abbildung 6: OOWS Kontexte Staff und Tutorship.....	- 30 -

Tabellenverzeichnis

Tabelle 1: Literatur zu den verschiedenen Modellen	- 12 -
Tabelle 2:Phasen von UWE	- 13 -
Tabelle 3: Literatur zur UWE	- 14 -
Tabelle 4:Phasen von WebML	- 15 -
Tabelle 5: Literatur zu WebML.....	- 17 -
Tabelle 6: Phasen von OOHDM.....	- 20 -
Tabelle 7: Literatur zu OOHDM.....	- 21 -
Tabelle 8: Weitere Methoden und Literaturhinweise	- 25 -

Glossar

Rich Internet Application	Ein Anwendung deren Logik- und Persistenzschicht auf einem Server im Internet ausgeführt wird. Die Präsentation der RIA findet sich im Browser. Die RIA weist Merkmale einer nativen Applikation auf. ¹
Unified Modeling Language (UML)	UML ist eine standardisierte Diagramm-Sprache zur Darstellung von Softwarearchitekturen und Geschäftsprozessen. Sie wird von der Object Management Group spezifiziert. ²
Hypertext	Hypertext beschreibt formatierten Text, der meist im Browser angezeigt wird. Hypertext enthält Referenzen zu anderen Texten, welches die vernetzte Struktur des Internets stützt.
Entity Relationship-Modell (ER-Modell)	ER-Modelle spiegeln physische Entitäten mit ihren Merkmalen und Beziehungen zueinander wieder. Damit helfen sie diese Entitäten in einer Datenbank abzubilden. ³
Hypermedia	Der Begriff Hypermedia erweitert die Definition von Hypertext. Dabei existieren hier auch verknüpfte Medien wie Bilder, Videos oder Audiodateien.
3-Schichten-Modell	Das 3-Schichten-Modell beschreibt eine weit verbreitete Softwarearchitektur, bei der der Aufbau der Anwendung in drei Ebenen

¹ Pfeil, C. (2009), S. 3

² Alhir, S. S. (1998), S. 7

³ Davis, W. S./Yen, D. C.(2010), S. 195

unterteilt ist. Es werden Präsentations-, Logik- und Persistenzschicht unterschieden.

XML

Die Extensible Markup Language beschreibt ein spezifiziertes Format zum Austausch von Daten. Es ist strukturiert und somit auch für Menschen gut lesbar.⁴

Use Case-Diagramm

Ein Use Case-Diagramm ist mit Hilfe der UML-Notation beschrieben. Es stellt einen bestimmten Anwendungsfall einer Software dar.

⁴ Kayser, S. (2007), S. 5: XML Dokumenttyp-Definition kurz und bündig, München: Grin Verlag

1. Einführung

Die Anforderungen und Vorgehensweisen der Entwicklung von webbasierten Anwendungen haben sich in den letzten Jahren stark verändert. Webanwendungen und insbesondere Oberflächen werden zunehmend komplexer wodurch mehr Aufwand in der Entwicklung solcher entsteht. Die neuen Anforderungen können durch die Bereitstellung funktionsstarker Entwicklungsbibliotheken verschiedener Technologien wie HTML5, JavaScript, AJAX oder Silverlight erfüllt werden. Dennoch werden beispielsweise Weboberflächen üblicherweise für eine bestimmte Technologie entwickelt und müssen bei einem Technologiewechsel mit entsprechendem Aufwand angepasst werden. Neben der Entwicklung einer Webanwendung anhand einer bestimmten Technologie, bestehen auch Ansätze welche den Fokus der modellgestützten Generierung von Webanwendungen im Fokus haben. Ziel ist dabei die technologieneutralen Aspekte einer Anwendung anhand von Modellen darzustellen und erst dann die Anwendung für eine konkrete technologische Basis zu generieren.

Die Methoden zur Vorgehensweise beim Entwickeln von Webanwendungen sind aus dem Software-Engineering hervorgegangen. Der Lebenszyklus der Software besteht aus mehreren Schritten welche von dem Zusammentragen der Anforderungsspezifikationen, der Anforderungsanalyse anhand von Use Cases, über den Entwurf der Software (UML) bis hin zur Implementierung und Validierung reichen. Web-Engineering baut auf diesen Methoden auf und erweitert diese durch Elemente welche die Charakteristika einer Webanwendung unterstützen.⁵ In den vergangenen Jahren ist eine Vielzahl an Ansätzen des UML-based Web Engineering wie zum Beispiel der Ansatz UWE entstanden, welcher im Wesentlichen eine Erweiterung der Unified Modeling Language (UML) darstellt.⁶

1.1. Problemstellung

Es gibt eine große Bandbreite an Ansätzen zur modellgestützten Generierung von Weboberflächen. Die Problemstellung der vorliegenden Arbeit liegt in einem fehlenden Überblick über die vorhandene Literatur zu den verschiedenen Ansätzen. Darüber hinaus ist eine Kategorisierung der Ansätze notwendig um besonders die Ansätze zu betonen, welche Modelle für die Generierung von Benutzungsoberflächen beinhalten. Modellbasierte Ansätze

⁵ Vgl. Morovona, T. (2008)

⁶ Vgl. Morovona, T. (2008)

zur Webentwicklung sind vor allem sehr theoretischer Natur und werden selten in der Praxis eingesetzt.

1.2. Ziele der Arbeit

Ziel der Arbeit ist es anhand einer Literatur-Recherche einen möglichst umfassenden Überblick über die aktuell bekannten theoretischen Ansätze und Methoden der modellbasierten Entwicklung von Webapplikationen zu schaffen. Das Ergebnis der Arbeit ist eine Zusammenfassung der vorhandenen Literatur mit einer Beschreibung welcher Ansatz in der Literatur behandelt wird und einer Bewertung inwiefern der theoretische Ansatz in der Praxis Anwendung findet.

Des Weiteren sollen die in der Literatur beschriebenen Ansätze nach ihrer Ausrichtung und Funktionalitäten untersucht und kategorisiert werden. Ein besonderer Schwerpunkt dieser Arbeit liegt auf der Darstellung der Ansätze und Methoden zur Generierung von Weboberflächen. Diese beinhalten sowohl die Präsentation der Oberfläche als auch die Navigation durch eine Webanwendung.

1.3. Methodik der Arbeit

Die verwendete Methodik dieser Arbeit ist eine Literatur-Recherche. Die den Autoren zugänglichen Quellen werden auf ihre Ansätze untersucht und miteinander verglichen. Des Weiteren werden die Ansätze anhand ausgewählter Kriterien kategorisiert und bewertet.

2. Einführung in das modellgestützte Web Engineering

Das World Wide Web hat sich in den vergangenen Jahren extrem rasant verändert. Relativ statische Webseiten, wie sie mit dem Web 1.0 in Verbindung gebracht werden entsprechen den Anforderungen der Nutzer seit langem nicht mehr. Von Webanwendungen des Web 2.0 werden eine hohe Interaktivität und sich dynamisch ändernde Inhalte erwartet. Die Inhalte werden hauptsächlich von den Nutzern selbst generiert⁷. Eine immer größer werdende Bedeutung kommt der Entwicklung des mobilen Webs zu. Mobile Endgeräte wie Smartphones, Tablets, Pocket PCs und weitere Handhelds finden eine schnelle Verbreitung. Mobile Webanwendungen bieten im Vergleich zu herkömmlichen Desktop Webanwendungen einige zusätzliche Features wie beispielsweise lokationsbezogene Services, sowie personalisierte und kontext-bezogenen Fähigkeiten⁸.

Eine Webanwendung ist im Vergleich zu einer traditionellen Softwareanwendung eine Entwicklungs- und Nutzungsplattform, welche auf den Technologien des Webs aufbaut. Eine Webanwendung wird nach Kappel definiert als: „[...] ein Softwaresystem, das auf Spezifikationen des World Wide Web Consortium (W3C) beruht und Web-spezifische Ressourcen wie Inhalte und Dienste bereitstellt, die über eine Benutzerschnittstelle, den Web-Browser, verwendet werden.“⁹

Die Hauptcharakteristika einer Webanwendung sind die Inhalte, die sich in kurzen Abständen ändern sowie der Umfang der Änderungen. Während die Inhalte einer Software relativ unverändert bleiben, werden Webanwendung beispielsweise durch die Einbindung des Nutzers ständig verändert. Neben den Inhalten ändern sich auch die Anforderungen an eine Webapplikation häufiger als bei einer traditionellen Software. Die Benutzungsoberfläche ist im Gegensatz zu traditioneller Software auf eine breitgefächerte Gruppe von Nutzern ausgelegt, welche verschiedene Erwartungen und Erfahrungen im Umgang mit Webapplikationen haben¹⁰. Die Besucherzahlen einer Webanwendung können extrem variieren.

Die zentralen Aspekte einer Webanwendung sind:

- Präsentation der Oberfläche / Benutzerinteraktion
- Navigation innerhalb der Anwendung
- Daten im Rahmen der Anwendung / Haltung und Verarbeitung (Inhalt)

⁷ Vgl. Murugesan, S. (2008), S.12 in: Rossi et. al. (2008)

⁸ Vgl. Murugesan, S.(2008), S.12 in: Rossi et. al. (2008)

⁹ Kappel, G. et.al. (2003), S.2

¹⁰ Vgl. Murugesan, S. (2008), S.8 in: Rossi et.al. (2008)

- Kommunikation mit anderen Komponenten/ Systemen (Workflow)

Weitere Problemstellungen von Webanwendungen werden von Grüne und Kneupner in einem Artikel des Magazins Wirtschaftsinformatik beschrieben. Dabei werden vor allem die Probleme durch unstrukturiertes Vorgehen bei der Entwicklung von Webanwendungen genannt.¹¹ Eine Problem sind beispielsweise die kurzen Versionszyklen von Webtechnologien, sowie die Anwendung von HTML Editoren welche zu einer „Quick and Dirty“-Vorgehensweise verleiten und die Präsentation vor die innere Struktur stellen.¹²

Die Entwicklung von Webapplikationen stützt sich auf die bekannten Methoden aus der traditionellen Softwareentwicklung. Aufgrund der spezifischen Charakteristika von Webanwendungen ist die Entwicklung allerdings komplexer und erfordert eigene Methoden des Web Engineering. Nach Murugesan wird das Web Engineering als eine Disziplin definiert, welche einen systematischen Ansatz verfolgt um die erfolgreiche Entwicklung, Einführung, und die Wartung eines Websystems- oder Webanwendung sicherzustellen.¹³

Aus der Wissenschaft des Model Driven Software Development (MDSO) heraus, haben sich unter dem Namen Model Driven Web Engineering (MDWE) einige Methoden und Ansätze zur modellbasierten Entwicklung von Webanwendungen entwickelt. Bei der Entwicklung von Webanwendungen ist ein solcher Ansatz insbesondere von Bedeutung da sich Web Technologien und Plattformen häufig weiterentwickeln.¹⁴ MDSO ist ein wichtiger Ansatz bei der Entwicklung komplexer, verteilter Systeme. In jeder Phase der Entwicklung eines Systems werden Modelle genutzt um einen Aspekt der Entwicklung wie beispielsweise die Präsentation jeweils unabhängig vom Rest des Systems zu entwickeln.¹⁵ Somit kann die Business Logik von der darunterliegenden technologischen Plattform abstrahiert werden. Durch Transformation zwischen den Modellen kann aus den einzelnen Modellen eine automatisierte Implementation des Systems erfolgen.¹⁶

In Abbildung 1 ist der schematische Aufbau der Webanwendung nach Anwendung der MDWE Methoden dargestellt. Zunächst werden Modelle der Struktur, Navigation, der Präsentation sowie der Benutzerschnittstellen generiert um diese dann in die verschiedenen Layer einer Webanwendung zu transformieren.

¹¹ Vgl. Grüne M./ Kneupner, R. (2002), S. 269

¹² Vgl. Grüne M./ Kneupner, R. (2002), S. 270

¹³ Vgl. Murugesan, S. (2008), S. 9 in: Rossi et.al. (2008)

¹⁴ Vgl. Kraus, A./Knapp, A./Koch, N. (o.J.)

¹⁵ Vgl. Moreno (2008) in: Rossi et. Al S.353

¹⁶ Vgl. Moreno (2008) in: Rossi et. Al S.354

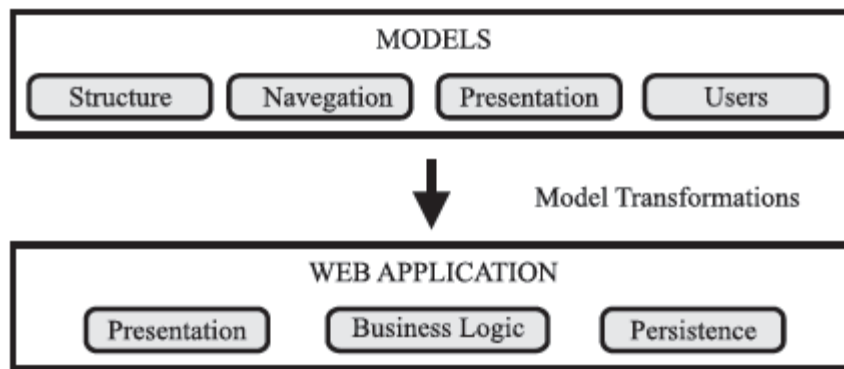


Abbildung 1: Das Schema der MDWE Methoden¹⁷

2.1. Was bedeutet MDA/ MDD

Der modellbasierte Ansatz zur Generierung von Software nennt sich Model Driven Architecture (MDA) und ist eine Ausprägung der MDSD, welche durch die Object Management Group propagiert wird.¹⁸ „MDA ist ein Vorschlag zur Lösung des Problems der Plattform unabhängigen Spezifikation von Anwendungslogik“.¹⁹

Neben dem Begriff der MDA hat sich die OMG Gruppe ebenfalls den Begriff des Model-Driven Development (MDD) eintragen lassen. MDD basiert wie auch andere Ansätze der Software Entwicklung auf Modellen. Die Modelle haben allerdings einen weitaus höheren Abstrahierungsgrad.²⁰ Ein Modell stellt eine vereinfachte Darstellung eines komplexen Systems dar und ermöglicht es ein System aus mehreren Sichten zu betrachten. Mit Hilfe einer Modellierungssprache und der Umsetzung von Regeln, können Modelle formgerecht aufgebaut werden. Die Regeln können wiederum in einem Modell dargestellt werden: dem sogenannte Metamodell, welches die Grundvoraussetzung für die automatisierte Transformation des Modells in Code darstellt.²¹

Ziel des Model Driven Development beziehungsweise der Model Driven Architecture ist es, ein System durch ein Modell von der Plattform abstrahiert und darzustellen. Dabei wird die Abhängigkeit einer spezifischen Technologie soweit wie möglich nach hinten geschoben.²² Durch Transformationen, kann ein plattformunabhängiges Modell in ein anderes plattformabhängigeres Modell überführt werden.

¹⁷ Vgl. Londoño, J./, Duitama, J.(2012), S.71

¹⁸ Vgl. Wolff (2011), S.7

¹⁹ Wolff (2011), S.7

²⁰ Vgl. Koch, N. (o.J), S.2

²¹ Vgl. Koch, N. (o.J), S.2

²² Vgl. Koch, N. (o.J), S.2

Durch die Trennung von Code und Modell, kann die Komplexität des Systems aufgehoben werden, Änderungen können eingepflegt, sowie Software wiederverwendet werden. In der Entwicklung von Webanwendungen findet MDD eine konkrete Anwendung. Für die modellgestützte Entwicklung von Webanwendungen wurden in den vergangenen Jahren eine Vielzahl an Modellierungssprachen vorgestellt, darunter Hera, OOHDM, OO-H, UWE, WebML, sowie W2000.²³ Als Beispiel für die Umsetzung des Model Driven Developments soll hier der UWE Prozess genannt werden, welcher unter Berücksichtigung der OMG Standards, den Regeln der MDA folgt.²⁴

Die Ansätze der modellgetriebenen Webentwicklung haben die Unified Modeling Language (UML) zu Grunde liegen, welche um einige Elemente erweitert wird, wie beispielsweise der Ansatz des UML-based Web Engineering (UWE).

2.2. Multikanalarchitekturen

Multikanalarchitekturen erfahren durch die steigende Anzahl mobiler webbasierter Zugänge eine immer größer werdende Bedeutung. Unter Multikanalarchitekturen versteht man verschiedene fachliche Zugänge (Nutzergruppen) sowie technische Zugänge zu einer Webseite.²⁵

Durch die Notwendigkeit einer Multikanalarchitektur müssen die Oberflächen einer Anwendung häufig umgeschrieben werden. Beispielsweise werden für die Benutzerfreundlichkeit einer Webseite Optionen für einige Nutzergruppen ausgegraut oder gar nicht erst angezeigt, während eine andere Nutzergruppe mehr Optionen hat. Neben den fachlichen Kanälen wie oben beschrieben, verbreitet sich die Nutzung unterschiedlicher technischer Kanäle immer weiter. Darunter fallen neue Technologien wie mobile Webanwendungen und die Spracherkennung.²⁶ Öffnet der Nutzer eine Webseite beispielsweise mit einem mobilen Endgerät wird idealerweise eine für kleine Bildschirme angepasste Seite geladen.

Kommt es zu inhaltlichen oder gar technologischen Änderungen einer Webanwendung so ist es immer erforderlich alle Oberflächen entsprechend anzupassen. Durch den Einsatz modellgestützter Webentwicklungsmethoden kann dies auf einer abstrahierten Ebene stattfinden und anschließend automatisiert der Code für die spezifischen Oberflächen generiert werden. Dadurch kann zum einen der Aufwand von Änderungen minimiert werden,

²³ Vgl. Koch, N. (o.J), S.2

²⁴ Vgl. Koch, N. (o.J), S.3

²⁵ Vgl. Hitz, M. (2013)

²⁶ Vgl. Hitz, M. (2013)

zum anderen wird die Fehleranfälligkeit beziehungsweise Inkonsistenz der Oberflächen reduziert.

3. Grundlagen und Besonderheiten der Modellierung von Web-Anwendungen

Die Grundlagen der Modellierung von Web-Anwendungen basieren auf denen der Softwareentwicklung. Hierbei wird zunächst ein Modell von der späteren Anwendung erstellt, um die relevanten Aspekte darzustellen. Allerdings ist die traditionelle Modellierung von Softwareanwendungen für Web-Anwendungen nicht ausreichend. Es gibt beispielsweise keine entsprechenden Konzepte, um die hypertextuellen Verlinkungen einer Web-Anwendung in einem Modell sinnvoll abbilden zu können.²⁷ Weiterhin muss für die Modellierung von Web-Anwendungen der Dokumentencharakter des Content, sowie die nichtlineare Navigation im Hypertext berücksichtigt werden. Daher wird bei der Modellierung von Web-Anwendungen in drei Ebenen, anstatt in zwei, unterteilt. Diese Ebenen sind auch die erste Dimension in der Abbildung 2, welche das Modell aufzeigt mit dem im Bereich Web Engineering gearbeitet wird. Dieses Modell ist in drei orthogonale Dimensionen aufgeteilt.

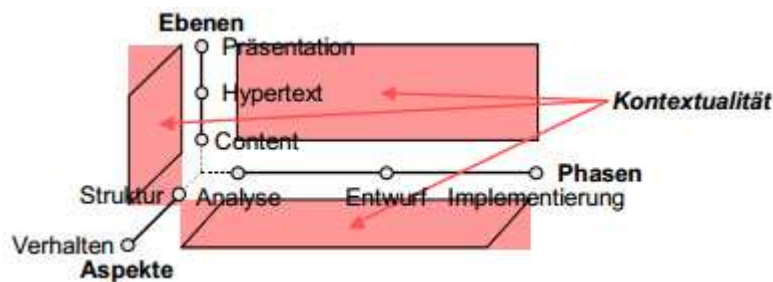


Abbildung 2: Anforderungen an die Modellierung von Web-Anwendungen²⁸

Die Ebenen der ersten Dimension sind:

- Content

Die Content Ebene entspricht den zugrunde liegenden Informationen der Web-Anwendung, sowie der Anwendungslogik. Das Ziel bei der Erstellung des Content-Modells ist die Festlegung der Informationsstruktur. Diese ist vergleichbar mit dem Datenbankschema in der Datenmodellierung.²⁹

²⁷ Vgl. Schwinger, W./ Koch, N. (2003)

²⁸ Vgl. Schwinger, W./ Koch, N. (2003)

²⁹ Vgl. Schwinger, W./ Koch, N. (2003)

- Hypertext

Die Hypertext Ebene zeigt die Gliederung des Contents in Knoten und Verweise zwischen diesen Knoten. Das Ziel der Hypertext-Modellierung ist es, die Navigierbarkeit durch den Inhalt der Web-Anwendung zu spezifizieren. Das Ergebnis ist zum einen ein Hypertext-Strukturmodell, welche die Struktur des Hypertexts definieren, als auch ein Zugriffsmodell, welches das Hypertext-Strukturmodell um Zugriffselemente erweitert.

- Präsentation

Die Präsentation Ebene ist die Gestaltung der Benutzerschnittstelle. Das Ziel dabei ist es eine einheitliche Präsentationsstruktur für einen Wiedererkennungseffekt beim Benutzer zu erzeugen, als auch die Struktur und das Verhalten der Benutzerschnittstelle so zu gestalten, dass die Interaktion mit der Web-Anwendungen für den Benutzer selbsterklärend ist.

Die Trennung in diese drei Ebenen hat einige Vorteile. Beispielsweise ermöglicht sie die Komplexität zu reduzieren und erlaubt eine Wiederverwendung.³⁰ So ist es möglich einen definierten Content für verschiedene Hypertextstrukturen und Zugriffsmodelle zu nutzen. Dieser Vorteil ist besonders nützlich, wenn es darum geht die Web-Anwendung für mobile Endgeräte oder für verschiedene Benutzergruppen zu erstellen. Ein umfassendes Modell einer Web-Anwendung beinhaltet alle drei aufgezeigten Dimensionen, allerdings kann es je nach Web-Anwendungen stark variieren.³¹

Kontextualität

In der Abbildung 2 ist neben den drei vorgestellten Ebenen ebenfalls die Kontextualität zu erkennen, welche die drei anderen Ebenen beeinflusst. Dabei beschreibt der Begriff der Kontextualität eine Anpassung der Webanwendung an den User oder den Inhalt den der User gerade betrachtet.³² Hier ist zum Beispiel das Ein- bzw. Ausblenden bestimmter Elemente aufgrund der Benutzergruppe gemeint. Der Aspekt der Kontextualität bezieht sich daher auf alle drei vorher genannten Ebenen und ist somit ganzheitlich bei der Modellierung der Webanwendung zu berücksichtigen. Durch die Anwendung kontextueller Modellierung kann dem Benutzer der Webanwendung eine personalisierte Oberfläche und eine auf dessen Intention zur Benutzung der Webanwendung angepasster Inhalt angezeigt werden.

³⁰ Vgl. Schwinger, W./ Koch, N. (2003)

³¹ Vgl. Schwinger, W./ Koch, N. (2003)

³² Vgl. Koch et. al. (2008), S. 159 in: Rossi, G. et.al. (2008)

4. Vorstellung der verschiedenen Methoden

In dem folgenden Kapitel werden die verschiedenen Modellierungsmethoden systematisch vorgestellt. Dabei geht es nicht darum, eine vollständige Aufzählung der aktuellen Methoden zu zeigen, sondern eine kompakte Übersicht. Die aktuellsten und meistverbreiteten Methoden werden tiefer gehend vorgestellt. Für eine weiterführende Recherche werden zudem weitere Quellen aufgezeigt.

4.1. Übersicht der verschiedenen Ansätze

Im Folgenden werden eine übersichtliche Auflistung aktueller Methoden, ihre zeitliche Einordnung und der Ursprung der jeweiligen Methoden dargestellt. Diese Ergebnisse wurden von zusammen von Nora Koch und Wieland Schwinger entwickelt.³³

Abbildung 3 zeigt diese Ergebnisse anhand einer Übersicht über die Modellierungsmethoden.

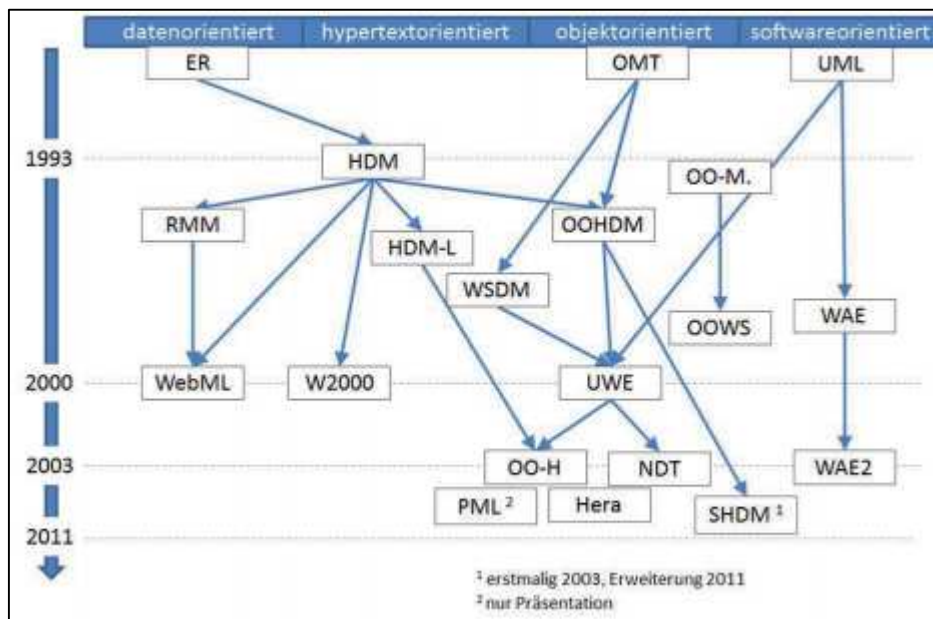


Abbildung 3: Übersicht über Modellierungsmethoden für Web-Anwendungen³⁴

Mit Hilfe dieser Abbildung ist zu erkennen, dass die verschiedenen Methoden alle auf drei Modelle zurückzuführen sind.³⁵

- Datenorientierte Methoden

³³ Vgl. Kappel, B. et.al. (2003)

³⁴ Vgl. Heinrich, A. / Gooren-Sieber, S. (2011)

³⁵ Vgl. Heinrich, A. / Gooren-Sieber, S. (2011)

Die datenorientierten Methoden basieren hauptsächlich auf dem ER-Modell und entspringen dem Bereich der Datenbanksysteme. Das ER-Modell wird zusätzlich um ein Konzept für die Modellierung von der Hypertext-Ebene erweitert.³⁶ Datenorientierte Methoden werden angewandt, um vor allem Web-Anwendungen zu entwickeln, welche datenbankgetrieben sind und ein hohes Transaktionsvolumen haben. Berühmte Vertreter sind dieser Methode sind das Relationship Management Methodology (RMM) und die Web Modeling Language (WebML), welche in dieser Arbeit noch detaillierter beschrieben wird.

- Hypertext-orientierte Methoden

Diese Methoden fokussieren sich auf den Hypertext-Charakter von Webanwendungen und haben sich hauptsächlich aus dem Bereich der Hypertext-Systeme entwickelt.³⁷ Beispiele für diese Methode ist das Hypertext Design Model (HDM), welches sich zu dem W2000 weiter entwickelte, sowie HDM-Lite und die Web Site Design Methode (WSDM).

- Objektorientierte Methoden

Die objektorientierten Methoden basieren entweder auf der Object Modeling Technique (OMT) oder auf UML, wobei gerade neue Methoden fast ausschließlich auf UML beruhen. Die Methoden Object-Oriented Hypermedia Design Method (OOHDM), UML-based Web Engineering (UWE) und die Object-Oriented Hypermedia Method (OO-H) sind die meistgenutzten Modelle für solche objektorientierten Methoden.

- Softwareorientierte Methoden

Die softwareorientierten Methoden betrachten Web-Anwendungen aus dem Blickwinkel der traditionellen Softwareentwicklung.³⁸ Daher ist es wenig überraschend, dass die Techniken hierfür die der klassischen Softwareentwicklung zu Grunde liegen. Beispiel hierfür ist das Web Application Extension (WAE) und die Weiterentwicklung WAE2.

Es ist ebenfalls zu erkennen, dass vor allem in den Jahren 1993 – 2000 viele Modellierungsmethoden entwickelt worden sind, jedoch die derzeitige Entwicklung deutlich zurückgegangen ist. Allerdings bedeutet dies nicht, dass nicht weiter an dem Thema geforscht wird, sondern viel mehr, dass die bereits vorhandenen Methoden weiter entwickelt und verfeinert werden.³⁹

³⁶ Vgl. Schwinger, W./Koch, N. (2003)

³⁷ Vgl. Schwinger, W./Koch, N. (2003)

³⁸ Vgl. Schwinger, W./Koch, N. (2003)

³⁹ Vgl. Heinrich, A. / Gooren-Sieber, S. (2011)

Eine weitere Klassifizierung ist in der Abbildung 4 von Schwinger und Koch dargestellt.

			Modellierungsmethode	Modellierungsparadigma	Notation	Weiterentwicklung	Anforderungsmoblie	Content-Moblie	Hyper	Präse	Kontextualitätsmodellierung	Struktur und Verhalten	Vorgehensweise	Werkzeugunterstützung	Generierung	Stärken
HDM-lite	HT	ER + eigene Notation	X	X	X	✓	✓	✓	X	s	eigene	Generierungswerkzeug	auto	Prozess zur Transformation der Modelle, automatische Generierung		
OO-H	OO	UML + eigene	✓	✓	✓	✓	X	pers	s + v	eigene	Modellierungs- & Generierungswerkzeug	auto	fortschrittliches Werkzeug für automatische Generierung			
OOHDM	OO	UML + eigene Notation	✓	✓	✓	✓	✓	pers	s + v	eigene	X	X	mächtige Konzepte für kontextuelle Navigation, Personalisierung			
RMM	DB	ER + eigene Notation	X	X	✓	✓	✓	X	s	eigene	Authoring-Werkzeug	semi	Hypertext-Modellierung basierend auf ER-Model, vordefinierter Prozess			
UWE	OO	UML	✓	✓	✓	✓	✓	pers	s + v	RUP	Modellierungs- & Generierungswerkzeug	semi	UML-basierte Methode, regelbasierte Personalisierung			
W2000 (HDM)	HT	UML	✓	✓	X	✓	✓	pers	s	X	Erweitertes UML-Werkzeug	X	benutzerzentrierte Hypertext-Modellierung			
WAE2 (WAE)	SW	UML	✓	✓	✓	X	✓	X	s + v	RUP	Standard-UML-Werkzeuge	X	Implementierungsentwurf, Architektorentwurf			
WebML	DB	ER, UML	✓	✓	✓	✓	X	pers	s + v	eigene	Modellierungs- & Generierungswerkzeug	auto	ausgereifte Notation, Datenbankintegration, Generierung			
WSDM	HT	eigene Notation	✓	X	✓	✓	X	X	s + v	eigene	X	X	benutzerzentrierte Vorgehensweise bei der Analyse			

✓	word unterstützt	pers	Personalisierung	RUP	Rational Unified Process	DB	datenorientiert
X	word nicht unterstützt					HT	Hypertext-orientiert
		s	Strukturmodellierung	eigene	eigene Vorgehensweise	OO	objektorientiert
		v	Verhaltensmodellierung	auto	automatische Generierung	SW	softwareorientiert
				semi	semiautomatische Generierung		

Abbildung 4: Übersicht und Kategorisierung der Ansätze⁴⁰

Der Fokus dieser Arbeit liegt, wie bereits erwähnt, nicht auf eine Auflistung aller Modellierungsmethoden, sondern ausgewählte Methoden auf ihre Methoden der Oberflächengenerierung und Navigation hin zu untersuchen. Aus den vorgestellten Klassifizierungen und Einteilungen werden daher die folgenden Methoden weiter fokussiert:

- UWE
- WebML
- OOHDM
- OOWS

Zudem werden für die Methoden RMMI, HDM-Lite, WSDM, OO-H, W2000 und WAE2 kurze Beschreibung, sowie weiterführende Literatur vorgestellt.

⁴⁰ Schwinger, W. / Koch, N. (2003), S.26

Weitere interessante Aspekte zum Vergleich der verschiedenen Methoden und ihren Ursprung sind die folgenden Literaturhinweise.

Titel	Autor
Hypermedia-Oriented Design An Approach for Supporting Evolvable Distributed Network Applications	Mike Amundsen
An Object-Oriented Design Method for Hypermedia Information Systems	Danny B. Lange
Web application development methods: a comparison	André L.S. Domingues, Sandro L. Bianchini, Marcella L.S. Costa, Fabiano C. Ferrari, José C. Maldonado
Designing Communication-intensive Web Applications: a Case Study	Vito Perrone, Davide Bolchini

Tabelle 1: Literatur zu den verschiedenen Modellen

4.2. Vorstellung ausgewählter Ansätze

4.2.1. UWE

UWE (UML-based Web Engineering) ist ein objektorientierter Ansatz für die systematische Entwicklung von Webanwendungen, welcher auf UML (Unified Modeling Language) aufbaut und als Erweiterung von UML gesehen werden kann. UML ist zwar eine sehr mächtige und praxistaugliche Modellierungssprache im Software Engineering Bereich, allerdings ist sie nicht ausreichend, um die Anforderungen zur Modellierung von Web Anwendungen erfüllen zu können. Mit UML ist es nicht möglich eine Hypertextstruktur abbilden zu können. Diese ist jedoch im Web Engineering elementar, da ansonsten keine Möglichkeit besteht die Navigation modellieren zu können. Zudem bietet UML keine Möglichkeit zur Modellierung der Oberflächen von Web Anwendungen.⁴¹ Hierbei ist vor allem der Begriff der Kontextualität zu erwähnen, welcher häufig auch als Adaption bezeichnet wird. Allerdings ist UML erweiterbar, so dass diese Anforderungen mit Hilfe von UWE, einer UML basierenden Modellierungssprache für Web Anwendungen erfüllt werden können.

⁴¹ Vgl. Schlimbach, C. / Schölzel, S.(2008)

UWE wurde erstmalig von Nora Koch im Jahre 2001 in ihrer Dissertation („Software Engineering for Adaptive Hypermedia Systems“)⁴² erwähnt. Daher ist es nicht überraschend, dass sämtliche Rechercheergebnisse in irgendeiner Weise mit Frau Koch zusammenhängen. Die offizielle Webseite, welche viele Informationen über UWE, inklusive einiger Tutorials bietet ist <http://uwe.pst.ifi.lmu.de/>.

Methodik des UWE Ansatzes

Der Ansatz der UWE Methodik unterteilt sich in den folgenden fünf Schritten der Modellierung:

Anforderungsmodell	Entspricht der Use Case Modellierung. Hierbei werden die gewünschten Funktionalitäten der späteren Anwendungen erfasst und dargestellt. Eine Besonderheit beim UWE ist, dass ebenfalls die Navigation zwischen dem Akteur und dem System modellierbar ist.
Konzeptuelles Modell	Modellierung der Anwendungsdomäne mit Berücksichtigung der jeweiligen Klassen und Assoziationen. Die Artefakte des Content Modells dienen der Hypertextstruktur.
Navigationsmodell	Das Navigation Model baut auf dem Content Model auf und ist für die Modellierung der Hypertextstruktur verantwortlich. Das Model wird nochmals in Navigation Space Model und Navigation Structure Model unterschieden.
Präsentationsmodell	Das Presentation Model beschreibt den strukturellen Aufbau der Benutzerschnittstelle und ist besonders wichtig für das Web Engineering. Dieser Aspekt wird nochmals gesondert im Kapitel 4 erläutert.
Prozessmodell	Modellierung von übergreifenden Prozessen der gesamten Anwendung, sowie Modellierung der im Navigation Modell definierten Geschäftsprozesse.

Tabelle 2:Phasen von UWE

Weitere interessante Aspekte und tiefere Erklärungen zu UWE sind die folgenden Literaturhinweise zu finden.

Titel	Autor	Kurze Beschreibung
Tutorials für UWE	LMU – Ludwig-Maximilians-Universität München	Tutorials für UWE

⁴² Vgl. Koch, N. (2003)

Modellbasierte Anforderungsanalyse für die Entwicklung von adaptiven RIAs	Sergej Kozuruba	Eine Diplomarbeit, welche die einzelnen Modelle von UWE sehr genau und gut erklärt.
UML-based Web Engineering. An Approach Based on Standards	Nora Koch, Alexander Knapp, Gefei Zhang, Hubert Baumeister	Das wohl bekannteste Paper über UWE, sehr detailliert und mit einem Beispiel erklärt. Jedoch ist der Artikel sehr tiefgehend und nicht für einen oberflächlichen Überblick gedacht.
UML-based Web Engineering	LMU – Ludwig-Maximilians-Universität München	Weiterführende Literaturliste und Projekte über UWE

Tabelle 3: Literatur zur UWE

Stärken und Kritikpunkte der UWE Modellierungsmethode

Eine Besonderheit von UWE, welche sich sehr stark von den anderen Methoden unterscheidet ist der Informationsgehalt und die Aktualität. UWE ist mit Abstand die am besten beschriebene Modellierungsmethode und es wird stetig daran weiter gearbeitet.

Eine weitere Stärke, insbesondere im Zusammenhang mit dieser Projektarbeit, ist die Möglichkeit der Adaption, welche besonders für die Multikanalarchitektur wichtig ist. Es ist möglich, bestimmte Kontextinformationen wie zum Beispiel Browsertyp, Endgerät oder Standort zu senden. Dadurch wird eine Multikanalarchitektur ermöglicht.

Ein weiterer Vorteil ist die Automatisierung. So ist es bereits möglich bestimmte Codeteile mit Unterstützung der jeweiligen Programme automatisch generieren zu lassen. Eine Liste der Programme ist hier zu finden: <http://uwe.pst.ifi.lmu.de/downloadsOverview.html>

Obwohl es viele Informationen über UWE gibt, ist die Einarbeitung in dieser Modellierungsmethode sehr aufwendig. Es ist sehr theoretisch und führt zu einer hohen Komplexität. Weiterhin ist die Praxistauglichkeit fraglich. Bisher gibt es keine nennenswerten praktischen Umsetzungsprojekte. Bisher ist UWE ein reines Forschungsprojekt und es konnten noch keine Best Practices entwickelt werden. Allerdings wird weiterhin daran gearbeitet und es ist durchaus denkbar, dass in naher Zukunft UWE in der Praxis eingesetzt werden kann.

4.2.2. WebML

WebML ist besonders für das Design von datenintensiven Web-Anwendungen geeignet. Seinen Ursprung hat WebML in dem datenorientierten Modellierungsmodell,

beziehungsweise dem Entity Relationship-Modell. Es wurde von Stefano Ceri und Marco Brambilla an der Polytechnischen Universität Mailand entwickelt und ist erstmalig im Jahr 2000 erschienen.⁴³ Die offizielle Internetseite ist <http://www.webml.org/> dort sind sehr viele Erklärungen, Tutorials und andere Materialien zu finden, welche WebML und den Umgang mit der Modellierung erklären.

Methodik von WebML

Im Folgenden sind die verschiedenen Phasen und Modelle von WebML dargestellt. Für ein besseres Verständnis sind die englischen Fachbegriffe nicht ins Deutsche übersetzt worden.

Modelle	Beschreibung des jeweiligen Modells
Data	Beim Data Model geht es darum, die Objekte der späteren Webanwendung zu definieren und die Zusammenhänge zwischen den einzelnen Objekten zu zeigen. Das Data Model ist mit einem Entity Relationship Model oder dem UML Klassendiagramm vergleichbar.
Derivation	Das Derivation Model ermöglicht es Objekte mehrfach anzeigen zu lassen, ohne sie mehrfach speichern zu müssen. Ermöglicht wird dies mit der WebML Object Query Language.
Composition	Das Composition Model bildet zusammen mit dem Navigation Model das Hypertext Model. In dem Composition Model wird definiert, welche Units die Page umfassen und welche Pages der Hypertext umfasst.
Navigation	Hierbei geht es darum die Navigation, anders ausgedrückt die Links, zu definieren.
Presentation	Modell zur Oberflächengestaltung.

Tabelle 4:Phasen von WebML

Die folgende Abbildung zeigt die Zusammenhänge zwischen den einzelnen Modellen.

⁴³Vgl. Heinrich, A. / Gooren-Sieber, S. (2011)

Data + Composition + Navigation + Presentation

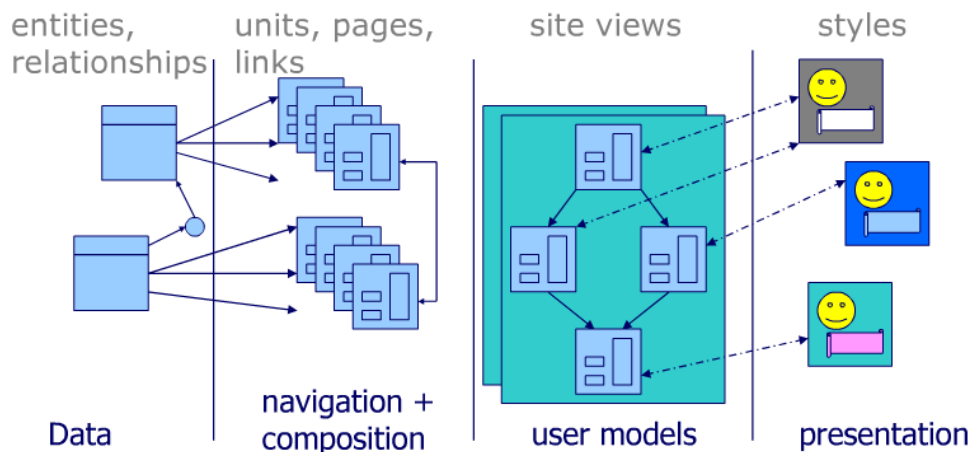


Abbildung 5: Zusammenhänge zwischen WebML Modellen⁴⁴

Content Units:

Zentrale Module der Sprache - ermöglichen die Datenanzeige und Dateneingabe. Z.B. die Data Unit, welche mit Hilfe von XML Informationen über ein einzelnes, bestimmtes Objekt darstellt. Neben dem Data Units gibt es noch viele andere Formen und Arten von Units.⁴⁵

Links – kontextabhängig oder nicht kontextabhängig. Die kontextabhängigen Links navigieren nicht nur zwischen den Units, sondern sie transportieren auch Informationen. Nicht kontextabhängige Links existieren zwischen Pages.⁴⁶

Site Views

Die Site Views stellen die Gesamtheit aus verschiedenen Pages da, welcher der User als eine gesamte Webseite sieht. Es können verschiedene Site Views erstellt werden, abhängig von den Usergruppen oder auch Geräten (mobile Endgeräte, Laptop).⁴⁷ Damit wird eine Zugriffskontrolle gewährleistet und die Seite wird automatisch auf das Endgerät angepasst, was wiederum für die Multichannel Architekturen sehr wichtig ist.

Stärken und Schwächen

Anhand dieser Abbildung ist bereits eine Schwäche zu erkennen. WebML ist sehr komplex und sehr aufwendig zu Beginn der Arbeit damit. Da alle Phasen aufeinander aufbauend sind, müssen auch alle entwickelt werden. Allerdings ist dies nicht für jede Web-Anwendung

⁴⁴ Brambilla, M. (2010)

⁴⁵ Vgl. Brambilla, M. (2010)

⁴⁶ Vgl. Brambilla, M. (2010)

⁴⁷ Vgl. Brambilla, M. (2010)

sinnvoll. Eine weitere Schwäche ist in Betracht der Thematik dieser Arbeit, dass WebML einen wenig umfassenden Ansatz zur Modellierung der Weboberfläche aufweist. Hierbei wird keine Beschreibung dieser angewandt, sondern auf die entsprechenden Tools verwiesen.

Weitere interessante Aspekte und Beispiele zu WebML sind die folgenden Literaturhinweise zu finden.

Titel	Autor	Beschreibung
Model-Driven Web Engineering (WebML)	ETH	Vorlesungsfolien von der ETH, welche einen guten Überblick über WebML geben.
Modellierung mit WebML	Business Informatics Group TU Wien	Vorlesungsfolien von der TU Wien, welche einen guten Überblick über WebML geben.
Designing Web Applications with WebML and Webratio	Marco Brambilla, Sara Comai, Piero Fraternali, Maristella Matera	Sehr detaillierte Beschreibung von WebML und die Umsetzung mit Web Ratio
Web Modeling Language (WebML): a modeling language for designing Web sites	Stefano Ceri, Piero Fraternali, Aldo Bongio	Beispielhafte Umsetzung von WebML, sehr technisch.

Tabelle 5: Literatur zu WebML

4.2.3. OOHDM

Object Oriented Hypermedia Design Method (OOHDM) ist eine Methode für das Design von komplexen Hypermedia-Anwendungen. Entwickelt wurde diese Methode von D. Schwabe, G. Rossi, C.J.P. Lucena und D.D. Cowan. Sie wurde erstmalig im Jahre 1995 vorgestellt.

Die Object Oriented Hypermedia Development Method stellt nicht nur eine Methodik bereit, die eine Trennung von Design, Inhalt, Struktur und Präsentation durch jeweilige Modelle umfasst, sondern auch eine klare Vorgehensweise zur Erstellung von Hypermedia. Um vom initialen Design bis hin zur Implementierung jeglichen Informationsbedarf bzw. Entwurfsschritt nachvollziehen und organisieren zu können folgt die schrittweise

Vorgehensweise verschiedenen Phasen. Diese Phasen werden je nach Autor der Quelle etwas unterschiedlich im Umfang beschrieben wobei auch die Anzahl leicht variieren kann. Je nach Quelle kann die Vorgehensweise in fünf Phasen aufgebaut sein oder auch nur in vier. Die Phasen die als Standard gelten sind die "Conceptual", "Navigational", "Abstract Interface Design" sowie die "Implementation".⁴⁸ Die Phase die, laut Quellen, eigentlich mehr eine Voraussetzung als ein Teil der Vorgehensweise ist, ist die "Requirements Gathering" Phase. All diese Phasen werden im nächsten Abschnitt genauer beschrieben.

Requirements Gathering	In dieser ersten Phase, oder auch Vorarbeit, geht es darum die Anforderungen der Nutzer und Auftraggeber zu sammeln und zu analysieren. Die Vorgehensweise in dieser Phase orientiert sich an den Nutzern. Es werden alle Nutzerrollen und Aufgaben, die diese Rollen erfüllen müssen, analysiert. Wenn die Nutzer und ihre Aufgaben bekannt sind, werden ganze Vorgehensweisen bzw. verschiedenen Szenarios erstellt und untersucht. Nachdem die Situation mit den Rollen der Nutzer und Aufgaben voll analysiert wurden, ist das Resultat ein Nutzer Interaktion Diagramm (User Interaction Diagramm - UID) welches genau diese Szenarios widerspiegelt und dabei den Fokus vor allem auf die Interaktion zwischen dem Nutzer und dem System legt.
------------------------	--

⁴⁸ Vgl. Dopatka, F. (2003)

Conceptual Design	<p>Beim "Conceptual Design" geht es, um das Konzeptmodell zur Beschreibung des Anwendungsgebietes mit Hilfe eines objektorientierten Konzepts aus Klassen, Attributen, Methoden und Sub-Systemen. Um daraus ein beschreibendes Diagramm der Internetanwendung zu machen fehlen noch Relationen zwischen den Objektklassen, welche mit einer Unified Modeling Language ähnlichen Notation beschrieben werden und in einem Entity Relationship – Diagramm resultieren. In dem Diagramm sind Klassen mit Attributen und Methoden, sowie Assoziationen, sprich Multiplizitäten, dargestellt. Weiter beinhaltet das Diagramm Informationen zu Vererbung und Aggregation.</p>
Navigational Design	<p>Die Klassen die in der "Conceptual Design" Phase erstellt wurden sowie alle zugehörigen Analysen der Datenstrukturen und Informationsflüsse und deren Niederschrift im Diagramm sind nun notwendig, um die Navigation zwischen den Code-Dokumenten zu definieren. Die Notwendigkeit besteht darin, zu analysieren wie die einzelnen Internetanwendungsbestandteile miteinander kommunizieren bzw. anwählbar und zugänglich sind.⁴⁹ Die Navigation richtet sich vor allem nach der Relation der Objekte und in welcher Hierarchie welche Anwendungsbestandteile aufgebaut sind.</p>

⁴⁹ Vgl. Koch, N. (1998)

Abstract Interface Design	In dieser Phase geht es nun um die Definition von wahrnehmbaren Objekten, sprich Bilder, Texte, Videos oder interaktive Objekte wie Karten. Hauptbestandteil sind daher Klassen für grafische Schnittstellen die Media Container enthalten sowie funktionelle Bestandteile wie Buttons oder anwählbare Menüs. Schnittstellen für die Interaktion mit dem Nutzer generieren somit wahrnehmbare Objekte die auf den analysierten Objekten vom "Navigational Design" beruhen. Externe und Nutzerevents müssen vom System wahrgenommen werden und es ist wichtig wie die Navigationsobjekte diese Events umsetzen und abarbeiten. ⁵⁰
Implementation	In der letzten Phase der Implementierung wird die genaue Umsetzung der Interface Objekte und des Navigationsverhaltens realisiert. Dabei werden konzeptionelle Entscheidungen getroffen, wie über die Client-Server Strukturen der einzelnen Anwendungsbestandteile oder auch über das grafische Design. Der Hauptbestandteil dieser Phase ist also die Generierung des Quellcodes und der Datenbankstruktur. Die Programmiersprache spielt bei OOHDM keine Rolle da diese Umsetzungsproblematik von vielen verschiedenen Sprachen gelöst werden kann. Somit gibt es also keine Einschränkungen in der Umsetzung so lange dem erarbeiteten Konzept gefolgt wird.

Tabelle 6: Phasen von OOHDM

Zusammenfassend ist zu sagen, dass sich diese Methode der Entwicklung von Internetanwendungen über das gesamte Spektrum von der Idee bis hin zur Implementierung bewegt und somit den Informationsbedarf von allen, an den Projekt beteiligten, Fachleuten abdeckt und durch die schrittweise Vorgehensweise einbezieht. Durch die Unabhängigkeit von Programmiersprachen ist diese Methode außerdem universell einsetzbar. Durch die

⁵⁰ Vgl. Schwabe, D./ Rossi, G./ Barbosa, S. (1996)

Trennung von Inhalten, Navigation durch die Anwendungsbestandteile und der konzeptionellen sowie programmiertechnischen Umsetzung ist einer sehr detaillierte Modellierung möglich. Dies macht die Methode allerdings auch sehr umfangreich und aufwendig, was eine klare Vorgehensweise fordert. Daher ist die Methode vor allem für große Projekte wie Internetshops oder umfangreiche Internetanwendungen prädestiniert, jedoch für kleiner eher statische Internetseiten viel zu komplex.

Weitere interessante Aspekte und Beispiele zu OOHDM sind die folgenden Literaturhinweise zu finden.

Titel	Autor	Beschreibung
The Object-Oriented Hypermedia Design Model (OOHDM)	Daniel Schwabe, Gustavo Rossi	Einstiegsdokument über OOHDM mit sehr vielen weiterführenden Links.
OOHDM Eine systematische Methode zum Entwurf von Web-Systemen	Nora Koch	Skript von Nora Koch zur oberflächlichen Vorstellung von OOHDM.

Tabelle 7: Literatur zu OOHDM

4.2.4. OOWS

Das Ziel der Object Oriented Web Solution (OOWS) Methode zur Entwicklung von Webapplikationen ist die Bereitstellung eines integrierten Frameworks. Dieses Framework soll die Entwicklungsprozesse zwischen dem Sammeln der Anforderungen an eine Webapplikation und der Erstellung eines lauffähigen Prototyps automatisieren. Dabei setzt OOWS auf eine Erweiterung der klassischen OO-Methoden des modellgestützten Web Engineerings. Diese klassischen Herangehensweisen umfassen lediglich die funktionale Analyse der Anwendung bis hin zu einer formalen Spezifikation. OOWS erweitert diese klassischen OO-Methoden um zwei weitere Modelle. Diese beiden Modelle bilden zusätzlich die Navigations- und Präsentationcharakteristika der zu erstellenden Webapplikation ab. Um diese Charakteristika zu identifizieren und zu spezifizieren sind zwei Schritte von besonderer Bedeutung – die Konzeptuelle Modellierung und die Entwicklung der Lösung.

Konzeptuelle Modellierung

Die Konzeptuelle Modellierung kann wiederum in drei weitere Schritte unterteilt werden. Im ersten und zweiten Schritt werden die funktionalen Anforderungen zusammengetragen und klassische Modellierungsaufgaben erfüllt. Das kann in Form von Use Case Diagrammen

(funktionale Analyse) oder ER- und Klassendiagrammen geschehen, welche als Werkzeuge der klassischen OO-Methode entspringen.⁵¹

Im dritten Schritt werden die Navigations- und Präsentationscharakteristika modelliert. Dabei werden auf der Basis der Klassendiagramme benutzerspezifische Navigationsgraphen erstellt. Deren Elemente und Navigationswege sind dann also auf die entsprechende Benutzergruppe abgestimmt. Die Beziehungen der einzelnen Elemente der Navigationsgraphen werden festgelegt und mit bestimmten Eigenschaften versehen.

Die so erstellten Navigationsgraphen bilden die Grundlage für die Modellierung der Präsentationscharakteristika. Hierbei werden die Elemente der Navigationsgraphen mit Eigenschaften versehen, die die Darstellung dieser bestimmen. Dies sind zum Beispiel die Bildung logischer Blöcke, die Reihenfolgen oder verschiedene Layouts.

Entwicklung der Lösung

Zur Entwicklung des Prototyps werden zuerst die Zielplattform und die Architekturdetails festgelegt. Unabhängig von der Wahl einer n- oder 3-Tier-Anwendung wird hier aufgrund des Fokus' der Arbeit nur die Generierung der Präsentationsschicht betrachtet. Bei der Kompilierung der Modelle werden die Elemente der Navigationsgraph und deren Eigenschaften in Webseiten umgewandelt. Die Webseiten sind mit einander verknüpft und bestehen aus einem Navigations- und einem Inhaltsbereich.

4.3. Literatur zu anderen Ansätzen

Die folgenden Quellen sind ebenfalls interessant für die Präsentationsmodellierung und sollten daher berücksichtigt werden. Allerdings sind deutlich unbekannter und somit konnten in der Recherche nur wenige Quellen und Informationen gefunden werden. Weiterhin würde das Beschreiben sämtlicher Methoden den Rahmen dieser Arbeit sprengen. Daher sind im Folgenden die Methoden kurz beschrieben und es wird eine weiterführenden Literatur zu dem Thema angegeben.

Web Engineering Methode	Beschreibung	Weitere Quellen
Web Site Design Method (WSDM)	Die WSDM Methode konzentriert sich vor allem auf die Benutzeransicht der Web-Anwendung. Der Ausgangspunkte sind hierbei die potentiellen Besucher der Web-	Model-driven web engineering methods: a literature review von Jesús Andrés Hincapié

⁵¹ Pelechano, V. et al. (o. J.)

	<p>Anwendung. WSDM ist in vier Phasen unterteilt:</p> <ol style="list-style-type: none"> 1. User Modeling 2. Conceptual Design 3. Implementation Design 4. Tatsächliche Implementierung 	<p>Londoño*, John Freddy Duitama</p>
<p>Object Oriented Hypermedia (OO-H)</p>	<p>Die OO-H Methode ist eine objektorientierte Methode, welche vor allem die spätere Benutzeransicht, die Zugriffsmöglichkeiten und die jeweiligen Navigationspfade modelliert. OO-H konzentriert sich vor allem auf die Personalisierung der Web-Anwendung, welche vor allem für eine Multichannel Architektur hilfreich ist. Obwohl in der Theorie OO-H eine besonders hilfreiche Methode wäre, um die Präsentationsschicht der Web-Anwendungen zu designen und zu erstellen, findet man in der Literatur nur wenig Anhaltspunkte darüber.</p>	<p>The OO-H Project - Web Engineering Group Model-driven web engineering methods: a literature review von Jesús Andrés Hincapié Londoño*, John Freddy Duitama</p>
<p>Web Software Architecture (WebSA)</p>	<p>Die WebSA ist eine modelbasierte Möglichkeit Web-Anwendungen zu entwickeln. Hierbei wird das Model der Web-Anwendung in drei Gesichtspunkte aufgeteilt:</p> <ul style="list-style-type: none"> • Anforderungen • Funktionalitäten • Architektonische Gesichtspunkte <p>WebSA basiert auf den Methoden UWE und OO-H.</p>	<p>Model-driven web engineering methods: a literature review von Jesús Andrés Hincapié Londoño*, John Freddy Duitama</p>
<p>Hypermedia Design Method Lite(HDM Lite)</p>	<p>Das HDM Modell berücksichtigt nur die Hypertext- und die Präsentationsschicht. HDM-Lite ist eine Weiterentwicklung von HDM, um besonders die Automatisierung von</p>	<p>Towards Modeling of DataWeb Applications - A Requirements' Perspective Werner Retschitzegger,</p>

	Hypertext- und Präsentationsschicht zu ermöglichen.	Wieland Schwinger
Hera	Hera ist eine sehr unbekannte Methode, welche sich jedoch auf die User- und Kontext Adaption fokussiert und daher einen sehr interessanten Ansatz darstellt. Hera ist abgeleitet von dem Hypertext –orientieren Ansatz, welche die Modellierung ebenfalls in die drei Phasen Content, Navigation und Presentation unterteilt. Eine weitere Besonderheit von Hera ist, dass es auf dem Resource Description Framework (RDF) basiert.	The Hera Research Program The Hera Research Program Modeling User Input and Hypermedia Dynamics in Hera von Geert-Jan Houben, Flavius Frascar, Peter Barna, and Richard Vdovjak A Model-driven Approach for Designing Distributed Web Information Systems von Richard Vdovjak and Geert-Jan Houben
Relationship Management Methodology (RMM)	RMM ist eine datenorientierte Modellierungssprache und der Vorgänger von WebML. Hierbei wird der Fokus vor allem auf die Beschreibung der Informationsobjekte, das heißt der Daten gelegt. RMM kann als eine Erweiterung eines ER Models gesehen werden, welches mit zusätzlichen Aspekte der Navigation versehen ist. RMM hat keine besonderen Aspekte zur Modellierung von Oberflächen. ⁵²	An RMM-Based Methodology for Hypermedia Presentation Design von Flavius Frascar, Geert Jan Houben, and Richard Vdovjak
W2000	W2000 ist eine Designmethode für beschreibende Modelle einer Webanwendung. Dabei stellt es sicher, dass die verschiedenen Aspekte einer Webanwendung in der Modellierung klar voneinander getrennt sind, um die Unabhängigkeit von der Technologie zur	W2000: A Modeling Notation for Complex Web Applications Luciano Baresi, Sebastiano Colazzo, Luca Mainetti, Sandro Morasca

⁵² Vgl. o.A. (o.J.)

	<p>späteren Implementierung zu wahren. Außerdem hilft W200 dabei, den Modellierungs- und Designprozess klar und strukturiert durchzuführen. Dieser Prozess folgt dabei vier vorgegebenen Phasen:</p> <ol style="list-style-type: none"> 1. Information Design zur Strukturierung der zu verarbeitenden Informationen 2. Navigation Design zur Definition der Nutzer-spezifischen Navigationspfade 3. Publishing Design zur Strukturierung des Inhalts in Pages und Publishing Units 4. Operations Design zur Modellierung von funktionalen und transaktionalen Elementen 	<p>An ontological approach to web application design using W2000 methodology von Anna Lisa Guido, Roberto Paiano und Andrea Pandurino.</p> <p>From Design to Development: a W2000-based Framework; Issues and Guidelines von Roberto Paiano und Andrea Pandurino.</p>
WAE / WAE2	<p>Die Methode der Web Application Extension und deren Weiterentwicklung WAE2 beschreiben eine Erweiterung von UML. Hier wurden der UML-Notation weitere Elemente zur Modellierung von Webanwendungen hinzugefügt. Dieser Ansatz unterstützt keine Generierung von Quellcode.</p>	<p>The Web Application Extension for UML Erwähnung in der Dissertation von Mihály Jakob: „Model-based Engineering of Web Applications: The flash WebMethod“, Seite 51</p>

Tabelle 8: Weitere Methoden und Literaturhinweise

5. Ansätze zur Generierung von Weboberflächen

Die Qualität einer Web-Anwendung hängt nicht nur von dem präsentierten Inhalt ab, sondern auch von einer qualitativ hochwertigen und benutzerfreundlichen Weboberfläche. Diese beinhaltet ebenfalls eine gut strukturierte Navigation.

5.1. Weboberflächen

Webanwendungen stellen interaktive Systeme dar, welche insbesondere von den Nutzern generierte Inhalte darstellen.

Die Schnittstelle zwischen dem System und dem Nutzer bildet die Weboberfläche inklusive der Navigation. Hier werden sowohl die konzeptuellen Inhalte der Webanwendung dargestellt, als auch die Funktionalitäten der Webanwendung zur Verfügung gestellt.

Um die Komplexität der Entwicklung zu minimieren werden bei den bekannten Modellierungsansätzen das Design der Weboberflächen sowie die Entwicklung der Funktionalitäten separat betrachtet.⁵³ Bei Webanwendungen sollte eine Trennung ebenfalls zwischen Oberflächen (presentation) und Navigation stattfinden, da Oberflächenereignisse nicht zwangsläufig eine Navigationsaktion auslösen.⁵⁴

Die Ansätze der modellbasierten Webentwicklung haben den Aspekt der Navigation als eigenständiges Feature erkannt und stellen Modelle und Notationen für dessen Entwicklung zur Verfügung.⁵⁵ Die Navigationsstruktur bestimmt welche Inhalte angesteuert werden können und welche Inhalte wiederum über einen anderen Inhalt erreicht werden können. Um die Funktionalität der Navigationsstruktur zu entwickeln werden Modelle genutzt, welche durch Abstraktionsmechanismen die unwichtigen Details vor dem Nutzer verstecken. Auf einem höheren Abstraktionsniveau kann die Navigation einer Anwendung in Form von Indices, Listen, Navigationsketten, etc. präziser geplant werden.⁵⁶

Ein abstraktes Präsentation- und Navigationsdesign ist besonders nützlich aufgrund der häufigen Technologiewechsel der Plattformen auf welcher Webanwendungen aufbauen. Eine weitere Anforderung an das Design von Webanwendungen hat die Einführung mobiler Endgeräte geschaffen. Einige Ansätze ermöglichen es Weboberflächen so zu entwickeln, dass diese unabhängig von der Darstellung auf dem Endgerät bleiben.⁵⁷

⁵³ Vgl. Rossi, G. et. al. (2008), S. 54 in: Rossi, G. et.al (2008)

⁵⁴ Vgl. Rossi, G. et. al. (2008), S. 54 in: Rossi, G. et.al (2008)

⁵⁵ Vgl. Rossi, G. et. al. (2008), S. 51 in: Rossi, G. et.al (2008)

⁵⁶ Vgl. Rossi, G. et. al. (2008), S. 52 in: Rossi, G. et.al (2008)

⁵⁷ Vgl. Rossi, G. et. al. (2008), S. 55 in: Rossi, G. et.al (2008)

Im Folgenden geht es darum die bereits eingeführten Modelle bezüglich der Generierung von Weboberflächen zu betrachten.

5.2. UWE

Wie bereits erläutert, wird bei der UWE Methode unterschieden zwischen dem Konzept Design, dem Navigation Design und dem Presentation Design. Im Folgenden soll vor allem die Erstellung einer Weboberfläche mit UWE untersucht werden. Dazu werden die UML Notationen und die UML Modelltechniken verwendet.

Bei dem Design der Präsentationsschicht wird ein so genanntes Abstract User Interface modelliert. Diese stellt dar, wie die Navigationsstruktur dem späteren User angezeigt wird. Folgende Aspekte müssen bei dem Design der Präsentationsschicht berücksichtigt werden:

- Graphische Anordnung der Navigationshierarchie
- Auswahl der Oberflächenobjekte
- Bestimmung der Schnittstellentransformation

Um diese Aspekte berücksichtigen zu können, wird zwischen einem statischen und dynamischen Präsentationsmodell unterschieden.

Das statische Präsentationsmodell wird mit einem UML Assoziations Diagramm erstellt, welches beschreibt wie die Benutzeroberfläche erstellt ist. Ein Objekt einer Benutzeroberfläche ist beispielsweise ein Text, ein Bild oder ein Button. Für die meist genutzten Objekte werden sogenannte Stereotypes definiert.

Bei dem dynamischen Präsentationsmodell hingegen wird das UML Sequenz Diagramm und das UML State Diagramm verwendet.

Genauere Informationen sind in der Publikation von Koch et al.⁵⁸ zu finden.

5.3. WebML

Über die Präsentationsmodellierung mit WebML liegen nur wenige Informationen vor, allerdings ist WebML eine der bekanntesten Modellierungsmethoden und daher werden die wenigen bisherigen Ansätze zur Präsentationsmodellierung erläutert:

Die Site Views stellen eine Gesamtheit aus verschiedenen Pages da, welche der User als eine gesamte Webseite sieht. Es können verschiedene Site Views erstellt werden, abhängig

⁵⁸ Extending UML to Model Navigation and Presentation in Web Applications Nora Koch, Hubert Baumeister, Rolf Hennicker and Luis Mandel Ab Seite 2-5

von den Usergruppen oder auch Geräten (mobile Endgeräte, Laptop). Damit wird eine Zugriffskontrolle gewährleistet und die Seite wird automatisch dem Endgerät angepasst, was wiederum für die Multichannel Architektur sehr wichtig ist.

Wie bereits erläutert besteht eine Site View aus verschiedenen Pages, welche der User als gesamte Webseite sieht. Eine Page ist immer mit einem oder mehreren Stylesheets verbunden, welche mit XML beschrieben werden. Die genaue Beschreibung der Oberfläche wird mit Hilfe des Programms Web Ratio erstellt. Somit gibt es im Gegensatz zu UWE keine formale Beschreibung der Oberfläche.

Ein gutes Paper zur Erstellung von Weboberflächen ist von Ceri et.al. geschrieben.⁵⁹

5.4.OOHDM

Wie bereits kurz erwähnt beziehen sich bei OOHDM vor allem das Navigational Design, sowie das Abstract Interface Design auf die Strukturierung der Weboberfläche. Diese beiden Design Phasen kombiniert werden dann in der Implementierungs-Phase, welche die letzte ist, durch eine Programmiersprache freier Wahl umgesetzt. Auch wenn die Navigational Design Phase sowie die Abstract Interface Design Phase zu gleich benötigt werden, da die Phasen aufeinander beruhen, konzentriert sich vor allem die Abstract Interface Phase auf die Interaktion und Kommunikation zwischen Nutzer und Web User Interface. In dieser Phase geht es um die Definition von wahrnehmbaren Objekten, sprich Bilder, Texte, Videos oder interaktive Objekte wie Karten. Hauptbestandteil sind unter anderem, wie bereits genannt, Klassen für grafische Schnittstellen die Media Container enthalten sowie funktionelle Bestandteile wie Buttons oder anwählbare Menüs. Es sollen somit wahrnehmbare Objekte die auf den analysierten Objekten vom "Navigational Design" beruhen generiert werden. Externe Events und Nutzer Events müssen vom System wahrgenommen werden und es ist wichtig wie die Navigation Objekte diese Events umsetzen und abarbeiten. Das Prinzip des Designs beruht hier auf der Darstellung von ADVs und ADOs. Eine ADV ist die Abstract Data View, die die Darstellung der Information, sprich die reine Präsentation, repräsentiert. Das ADO, also das Abstract Data Object, stellt die zu präsentierenden Daten aus dem Navigational Design und dem Datenmodell bereit. Das heißt ein ADO kann mehrere ADV besitzen, da eine bestimmte Information, also Daten aus dem Datenmodell, auf verscheiden Weise auf der Website dargestellt werden können. Die externen sowie auch Kunden Events werden zusammen mit den ADVs und ADOs unter Verwendung einer vorgegebenen Notation als schematisches Bild dargestellt. Durch ein weitere Schema, das ADV-Chart,

⁵⁹ Model-driven Development of Context-Aware Web Applications von Stefano Ceri, Florian Daniel, Maristella Matera, Federico M. Facca Ab Seite 9

werden abschließend noch die Interaktionen und Abhängigkeiten der einzelnen ADVs beschrieben, sprich bestimmte Events die dazu führen das ADV gezeigt oder verborgen werden sowie Reaktionen von ADVs untereinander. Das ADV-ADO Schema sowie das ADV Chart wird dann von Web-Designer sowie Datenbankmodellierern verwendet um in der Implementierungsphase die Codestruktur zu generieren und die Weboberfläche final um zu setzen.

5.5.OOWS

OOWS erweitert klassische OO-Methoden der modellgestützten Entwicklung von Webapplikationen um Aspekte der Generierung der Präsentationsschicht. Dabei werden weitere Modelle eingeführt, welche die Generierung von Navigations- und Präsentationselementen unterstützen. Diese sind für die Entwicklung der Navigations- und Präsentationscharakteristika anzuwenden. Nach dem Schritt der Modellierung werden die Modelle durch eine Kompilierung in einem Prototyp übertragen.

Modellierungen der Navigations- und Präsentationscharakteristika

Nachdem die Analyse der funktionalen Anforderungen an die Webapplikation und die klassische OO-Modellierung abgeschlossen sind, wird bei der Anwendung von OOWS die Modellierung der Navigations- und Präsentationscharakteristika vorgenommen.

Im Zuge der der Navigationsmodellierung wird die Struktur der späteren Webapplikation entworfen und an die verschiedenen Benutzergruppen angepasst. Dazu wird für jede Benutzergruppe ein Navigationsgraph angelegt. Diese spezifiziert die Strukturen, auf die eine Benutzergruppe zugreifen kann und darf, sowie die Benutzergruppe selbst. Angenommen es gibt n verschiedene Benutzergruppen, so gäbe es auch n verschiedene Navigationsgraphen. Die Navigationsgraphen bestehen aus Knoten und Kanten. Dabei stellen Knoten einen Kontext und Kanten eine Beziehung zwischen mehreren Kontexten dar. Ein Kontext beschreibt alle Attribute und Operationen mehrerer Klassen (abgeleitet aus dem Klassendiagramm), die für einen bestimmten Benutzer sichtbar bzw. verwendbar sind. Der Start der Navigation bzw. des Graphen wird durch eine Manager-Klasse dargestellt. Außerdem können zusätzliche, optionale Informationen über bestimmte Beziehungen an eine andere Klasse gebunden werden. Die Relationen sind dabei immer bidirektional und können verschiedener Arten sein (Assoziation, Komposition, Aggregation, Spezialisierung oder Generalisierung). Der Kontext kann außerdem mit weiteren Eigenschaften versehen werden. So können Attribute Indizes zur Darstellung in Listen erhalten oder Filter für Klassen anhand von Attributen festgelegt werden, mit Hilfe derer Suchen durchgeführt werden

können. Die Abbildung 6 zeigt die zwei Kontexte Staff und Tutorship mit den verschiedenen Klassen (<<views>>), deren Beziehungen und weiteren Eigenschaften für den Kontext Staff.

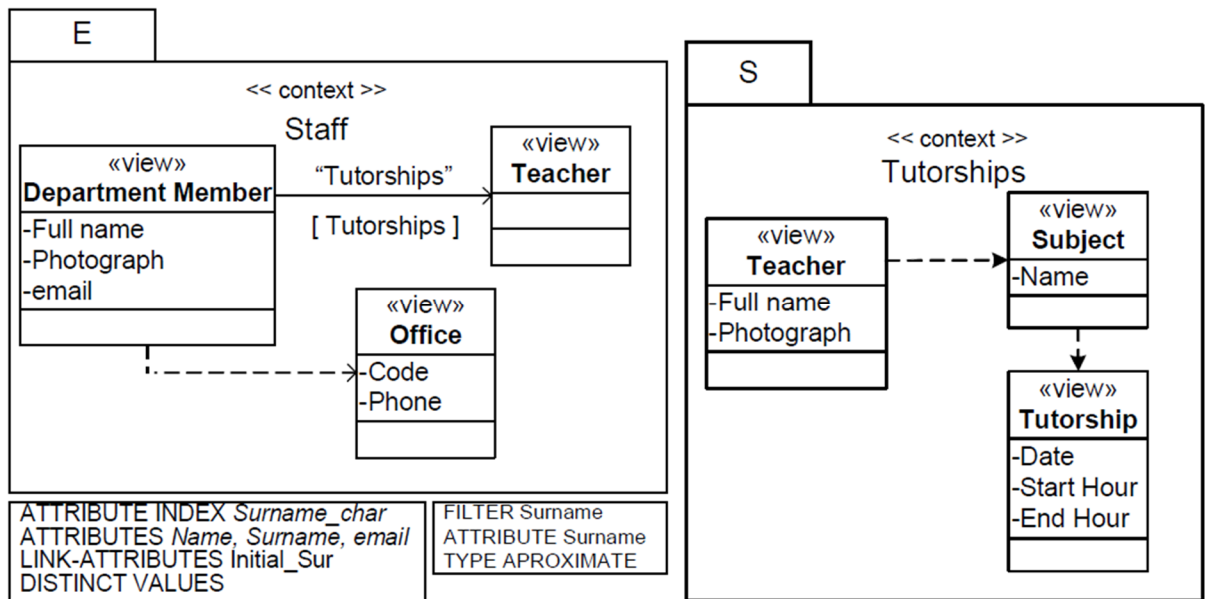


Abbildung 6: OOWS Kontexte Staff und Tutorship

Als Grundlage für die Modellierung der Präsentation werden die Ergebnisse der Navigationsmodellierung herangezogen. Dabei werden die Kontexte als Basisobjekte der Modellierung gewählt. Die Eigenschaften der Darstellung werden dann den Objekten wie Klassen, Beziehungen oder Attributen zugewiesen. Dabei können drei verschiedene Eigenschaften unterschieden werden. Zum einen werden logische Einheiten erstellt und voneinander getrennt. So können zum Beispiel zwei Instanzen einer Klasse als zwei logische Einheiten auf zwei Webseiten angezeigt werden. Zwischen diesen Webseiten kann über eine weitere Eigenschaft eine geordnete Navigation angelegt werden. Die Ordnung kann auf einer bestimmten Reihenfolge oder zum Beispiel einer Sortierung anhand eines Attributes beruhen. Mit der dritten Eigenschaft lässt sich das Layout der darzustellenden Informationen festlegen.

Kompilierung der Modelle

Nachdem die Navigation und die Präsentation modelliert wurden, wird mit Hilfe eines Kompilers ein Gerüst der Webapplikation erzeugt. Dieses besteht aus miteinander verlinkten Webseiten, die je nach Benutzergruppe verschiedene Kontexte widerspiegeln. Bei der Überführung der Modelle werden so die Knoten der Navigationsgraphen in Webseiten umgewandelt. Die Verknüpfung mit anderen Webseiten erfolgt über die Interpretation der Kanten. Anhand der Definition eines Kontextes werden die entsprechenden Informationen

angezeigt. Die kompilierte Webanwendung besteht dann also mit einander verknüpften Webseiten, welche aus einem Navigation- und einem Inhaltsbereich bestehen.

5.6. XUL und XFORMS

XUL

XUL ist eine XML basierende Anwendung zum Erstellen von User Interfaces. Daher auch die Abkürzung XUL – XML User Interface Language, welche Mozilla für das Projekt einer neuen User Interface Description Language verwendet, die vor allem von Mozilla Browsern unterstützt wird wie zum Beispiel Firefox. Ziel der neuen Description Language für User Interfaces sind vergleichbar umfangreiche Widgets und Oberflächengestaltung wie in Swing Frameworks und anderen Toolkits. Das Grundprinzip basiert auf der Syntax von XHTML, auch wenn XUL und XHTML nicht vollständig kompatibel sind, sprich XHTML Spezifikationen Tag-Beschreibungen stimmen nicht mit den Spezifikationen von XUL überein. Daher ist es auch nicht verwunderlich, dass XUL kein offizieller Standard ist und kein offizielles Schema oder DTD (Document Type Definition) existiert. Alle XUL Spezifikationen sind ausschließlich auf der offiziellen XUL Documentation Website beschrieben.⁶⁰

Mozilla entwickelte XUL vor allem um eine Lücke zwischen Programmierumgebungen, wie zum Beispiel Swing, SWT, AWT oder MFC, und Web Markup-Sprachen, wie HTML, CSS, DHTML oder Java Script, zu schließen. Der Sinn findet sich darin, die Komplexität (lange Entwicklungszeiten, hoher Lernaufwand) und somit die Schwierigkeit der Frameworks mit XUL zu umgehen ohne dabei auf deren umfangreiche Oberflächen-Gestaltungsmöglichkeiten zu verzichten. Markup-Sprachen sind einfacher Strukturiert, aber auch wesentlich eingeschränkter was die Struktur und Gestaltung des User Interface angeht. XUL soll also die Simplizität der Markup-Sprachen adaptieren und gleichzeitig die umfangreichen Strukturmöglichkeiten der User Interface bieten, die auch von den Frameworks zur Verfügung gestellt werden können. Um dies zu erreichen nutzt XUL Markup um das Graphical User Interface schnell und umfangreich zu gestalten sowie durch das Einbinden von Java Script basierenden Widgets die den Gesamteindruck der Web Oberfläche grafisch und funktionell aufwerten. Auch wenn die Struktur von XUL sehr der von HTML ähnelt, wird das HTML basierende Seiten Layout für die Struktur der Seite nicht umgesetzt sondern durch eigenen Verfahren realisiert. Detailliertere Beschreibungen zu Umsetzungen von XUL werden in dieser Ausarbeitung nicht dargestellt, da dies den Rahmen der Literaturrecherche sprengen würde und daher auf die genannten Quellen verwiesen wird.

⁶⁰ <https://developer.mozilla.org/en-US/docs/XUL>

Abschließend zu der kurzen Beschreibung von XUL, wie sie in der wenigen Literatur zu XUL zu finden ist, werden zusammenfassend noch einmal die Vorteile, aber auch die Bedenken zu dem Mozilla Projekt dargestellt, ebenso basierend auf Literatur Quellen wie der offiziellen Mozilla XUL Definition Website und einer Ausarbeitung von dem Programmierer Eitan Suez.⁶¹

Vorteile

XUL ermöglicht eine schnelle, unkomplizierte Entwicklung und Erstellung von grafisch und funktionell ansprechenden und aufwendigen Web-Oberflächen die als Interface die Usability für den Nutzer stark erhöhen. Durch die bereits sehr ausgeprägte Kompatibilität mit CSS, Java Script oder DOM werden bereits geläufige und verbreitete Markup-Sprachen für die User Interface Gestaltung wiederverwendet und somit auf vorhandenes Wissen zurückgegriffen. Durch die einfache Strukturierung des XUL Codes sind saubere, leserliche und vor allem wartbare Front-End Entwicklungen möglich. Aufgrund dieser Vorteile, sowie der Einzigartigkeit des Prinzips von XUL, wird weiter an einer Standardisierung von XUL gearbeitet. Da jedoch Mozilla dafür bestimmte Rechte zum Adaptieren von formalen Richtlinien und Standards fehlen, ist unklar wann und ob die Standardisierung vollzogen werden kann.

Bedenken

Anknüpfend an das letzte Statement in den Vorteilen ist mit das größte Bedenken, dass XUL kein offizieller Standard ist und somit ein ausgereiftes Schema für die Verwendung sowie eine umfangreiche Dokumentation fehlen. Bei einer genaueren Betrachtung des XUL Beispiele und der Entwicklungsumgebung für XUL wird klar das sich XUL sehr auf lokale Anwendungen beschränkt, sprich Web Anwendungen die lokal installiert sein müssen, was laut Literatur Recherchen, dem Prinzip von Web Anwendungen entgegen wirkt, da normalerweise da Ziel ist diese eben nicht lokal sondern über Netzwerke (Internet oder Intranet) bereit zu stellen. Außerdem ist zu Sagen das sich in der Entwicklung von XUL seit Jahren nicht viel getan hat, was vor allem am Alter der zu findenden Literatur oder Forenbeiträge zu erkennen ist. Es ist also mit Vorsicht zu betrachten wenn auf XUL referenziert wird oder auch zugehörige Toolkits wie LUXOR.⁶²

XFORMS

⁶¹ Developing GUIs with XUL, Eitan Suez, www.uptodata.com

⁶² <http://luxor-xul.sourceforge.net/>

Da XFORMS ein W3C Standard ist und somit über eine ausführliche Beschreibung sowie Definition und Dokumentation verfügt, geht es in diesem Abschnitt rein um die kurze Vorstellung von XFORMS und die Idee hinter diesem Konzept. Jegliche Informationen in diesem Abschnitt beziehen sich auf die offizielle "W3C Recommendation"⁶³ und eine Ausarbeitung von Christoph Hofmann und Gregor Liebermann⁶⁴ sowie der offiziellen Information-Website von W3C zu Forms.⁶⁵

Die Idee von XFORMS beruht auf der Tatsache das heutige Websites, sobald es zu einer Interaktion mit dem Nutzer kommen soll, auf Forms basieren die die Möglichkeit der Eingabe von Daten und Auswahl von gegebenen Optionen bieten. Forms sind eine Art Formularumgebung die in einer HTML basierende Oberflächenstruktur der Internetseite die Möglichkeit geben mit dem Nutzer über Eingaben zu interagieren. Das heißt Daten können vom Nutzer über die grafische Oberfläche abgefragt werden und auf dem Server verarbeitet werden wonach dann die Reaktion des Servers und somit der Website und die Informationen für den Nutzer generiert werden. Auf dieser Basis lässt sich schlussfolgern, dass, je komplexer die Interaktion der Website mit dem Nutzer gestaltet ist, desto umfangreicher die Form-Struktur aufgebaut sein muss. In der Zeit des Internetverkaufs sowie Internetanwendungen nehmen diese Form-Strukturen somit kaum zu überblickende Ausmaße an. Zu der wachsenden Komplexität kommt die Vielfalt an Geräten die zum Abrufen dieser Forms genutzt werden, sprich nicht nur PC Systeme, sondern auch mobile Endgeräte wie Tablets oder Smartphones. Eine notwendige Reaktion war die Erweiterung bzw. Erneuerung von Forms durch individuelle Elemente die die Syntax von Forms erheblich vereinfachen und optimieren mit den Zielen der Wiederverwendbarkeit, Plattformunabhängigkeit und universellem Einsatz in verschiedenen Web Anwendungen. Der Das Konzept und die Verwendung von XFORMS ist somit der von traditionellen FORMS absolut ähnlich wenn nicht sogar gleich, die Idee ist Forms zu optimieren und funktioneller zu gestalten. Darum geht es mehr darum die individuellen Elemente von XFORMS zu verstehen und das Konzept als die allgemeine Verwendungsweise. Der Hauptunterschied liegt somit in den Elementen sowie im Konzept, das bei XFORMS auf MVC (Model View Controller) basiert. Grund ist die strikte Trennung vom Graphical User Interface und den zu verarbeitenden Daten, was eine Grundvoraussetzung für die Plattformunabhängigkeit ist. Die Präsentation ist somit weiter durch XHTML dargestellt und mit XFORM Elementen, die die Form Funktionalitäten erweitern, gefüllt während die Daten durch XML plattformunabhängig transportiert und verarbeitet werden können, was eine Kommunikation von Server und Client

⁶³ W3C Recommendation 14 October 2003

⁶⁴ XForms - The Next Generation of Internet Interfaces, Augsburg University of Applied Sciences

⁶⁵ <http://www.w3.org/MarkUp/Forms/>

Endgerät unabhängig machen und sogar zwischen verschiedenen Anwendungen ermöglichen. Detailliertere Beschreibungen und Umsetzungen können in den genannten Literaturen gefunden werden, sprengen jedoch den Umfang einer Literaturrecherche.

6. Zusammenfassung und Literaturübersicht

Im folgenden Kapitel werden die Themen anhand einer Literaturübersicht zusammengefasst. Eine besonders gute Übersicht über die Ansätze des MDWE ist in der Literatur von Rossi et. al. zusammengefasst. Im Jahr 2008 ist eine Sammlung verschiedener Beiträge in der Quelle „Web Engineering- Modeling and Implementing Web Applications“ veröffentlicht worden. Insbesondere der zweite Teil des Buches unter der Überschrift „Web Design Methods“ ist in die Ausarbeitung der vorliegenden Arbeit in großem Maße eingeflossen.

Besonders bekannt für Publikationen bezüglich des modelbasierten Web Engineerings ist der Fachkreis MDWEnet, welcher seit 2005 jährlich Workshops abhält. Der Fachkreis ist aus einer Initiative des ISWE (International Society for Web Engineering) heraus entstanden. Zum Lenkungskreis zählen Antonio Vallecillo, Gustatvo Rossi, Geert-Jan Houben sowie Nora Koch. Unter der Leitung von Nora Koch sind in den vergangenen Jahren einige Diplom- sowie Doktorarbeiten zum Thema MDWE entstanden. In der Publikation aus dem Jahr 2007 stellt sich die MDWEnet Initiative vor und stellt erste Ergebnisse dar. Dabei geht es um die Interoperabilität der verschiedenen Ansätze. Jeder Ansatz stelle verschiedene Modelle (Navigation, Präsentation, Data, etc.) zur Verfügung, sowie die dazugehörigen Compiler, welcher die Weboberfläche sowie die darunterliegende Logik generiert. Dennoch weisen die Ansätze nach Ansicht der MDWEnet Initiative einige Limitationen auf. Diese Limitationen können durch die Interoperabilität der verschiedenen Ansätze bzw. Modelle behoben werden. Die Modelle der Ansätze sollen demnach komplementär verwendet werden und sich gegenseitig ergänzen.

Neben der oben genannten Hauptpublikation zum Thema model driven Web Engineering wurden weitere Quellen aufgefunden. In Tabelle 9 wird ein Überblick über die Literatur gegeben. Es wird der Autor genannt sowie eine Beschreibung geliefert auf welchen Ansatz sich die jeweilige Publikation fokussiert.

In der Dissertation von Wolff (2011) wird die Thematik der modellbasierten Generierung von Benutzungsoberflächen allgemein in Betracht genommen. Dabei werden vor Modellierungstechniken für Softwaresysteme sowie Modelle für die Nutzerinteraktion (User Interfaces), sowie deren Entwicklung näher erläutert.

Die im Januar 2013 in der Fachzeitschrift „International Journal of Innovative Computing, Information and Control“ , erschienene Publikation von Aragon et.al. beinhaltet eine Auswahl der wichtigsten MDWE Entwicklungsansätze, welche analysiert und von Kriterien aus der Literatur miteinander verglichen werden. Die Publikation kommt zu dem Ergebnis,

dass der neue Trend der MDWE Ansätze einen neuen Weg für praxisnahe Projekte eröffnet und die Lücken der klassischen Softwareentwicklung schließen kann.

Allgemeine Literatur zu MDWE:

<u>Titel:</u> A Practical Approach to Achieving Interoperability of Model-Driven Web Engineering Methods
<u>Autor:</u> Antonio Vallecillo, Nora Koch, et.al.
<u>Beschreibung:</u> Veröffentlichung des Arbeitskreise MDWEnet. Besonderer Fokus auf die Interoperabilität zwischen den Ansätzen des MDWE.
<u>Quelle:</u> http://www.wit.at/people/schauerhuber/publications/mdwenet.pdf

<u>Titel:</u> Web Engineering: Modeling and Implementing Web Applications (Human-Computer Interaction Series)
<u>Autor:</u> Gustavo Rossi, Oscar Pastor , Daniel Schwabe, Luis Olsina (Herausgeber)
<u>Beschreibung:</u> Veröffentlichung des Arbeitskreises MDWE- Zusammenfassung der wichtigsten Publikationen aus 2008
<u>Quelle:</u> Universitätsbibliothek Stuttgart

<u>Titel:</u> An analysis of model-driven web engineering Methodologies
<u>Autor:</u> Andreas Wolff (2011)
<u>Beschreibung:</u> Dissertation an der Universitat Rostock mit einem besonderern Bezug auf Benutzungsoberflächen
<u>Quelle:</u> http://rosdok.uni-rostock.de/file/rosdok_derivate_000000004659/Dissertation_Wolff_2011.pdf

<u>Titel:</u> An analysis of model-driven web engineering Methodologies
<u>Autor:</u> Kraus, A./Knapp, A./Koch, N. (o.J.)
<u>Beschreibung:</u> Ein guter Überblick über MDW.

Quelle: <http://aran.library.nuigalway.ie/xmlui/handle/10379/3414> International Journal of Innovative Computing, Information and Control Volume 9, Number 1, January 2013

Allgemeine Literatur zu den Methoden:

Titel: Hypermedia-Oriented Design An Approach for Supporting Evolvable Distributed Network Applications

Autor: Mike Amundsen

Beschreibung: Sehr detailliertes Paper zu den Hypertextorientierten Methoden.

Quelle: <http://www.w3.org/2011/10/integration-workshop/p/hypermedia-oriented-design.pdf>

Titel: An Object-Oriented Design Method for Hypermedia Information Systems

Autor: Danny B. Lange

Beschreibung: IBM Research Paper, welches erläutert warum Hypertext Modellierungsmethoden besser sind für die Modellierung von Web-Anwendungen und nur der objektorientierte Ansatz nicht ausreichend ist.

Quelle: <http://www.moe-lange.com/danny/docs/hicss27.pdf>

Titel: Web application development methods: a comparison

Autor: André L.S. Domingues, Sandro L. Bianchini, Marcella L.S. Costa, Fabiano C. Ferrari, José C. Maldonado

Beschreibung: Eine der wenigen Papers, welche die bekanntesten Methoden vergleicht.

Quelle: http://www2.dc.ufscar.br/~fabiano/pub_arquivos/Domingues_Webmedia2007.pdf

Titel: Designing Communication-intensive Web Applications: a Case Study

Autor: Vito Perrone und Davide Bolchini

Beschreibung: Paper über ausgewählte Methoden und Designmethoden für Web-Anwendungen mit einem hohen Kommunikationsgrad.

Quelle: http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER04/Vito_Perrone.pdf

Allgemeine Literatur zu UWE:

<u>Titel:</u> Model Driven Generation of Web Applications in UWE
<u>Autor:</u> Kraus, A./Knapp, A./Koch, N. (o.J.)
<u>Beschreibung:</u> Ein guter Überblick über UWE
<u>Quelle:</u> http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.66.9883

<u>Titel:</u> Tutorials für UWE
<u>Autor:</u> LMU – Ludwig-Maximilians-Universität München
<u>Beschreibung:</u> Tutorials für UWE
<u>Quelle:</u> http://uwe.pst.ifi.lmu.de/

<u>Titel:</u> Modellbasierte Anforderungsanalyse für die Entwicklung von adaptiven RIAs
<u>Autor:</u> Sergej Kozuruba
<u>Beschreibung:</u> Eine Diplomarbeit, welche die einzelnen Modelle von UWE sehr genau und gut erklärt
<u>Quelle:</u> http://uwe.pst.ifi.lmu.de/publications/KozurubaDA.pdf

<u>Titel:</u> Uml-based Web Engineering An Approach Based on Standards
<u>Autor:</u> Nora Koch, Alexander Knapp, Gefei Zhang, Hubert Baumeister
<u>Beschreibung:</u> Das wohl bekannteste Paper über UWE, sehr detailliert und mit einem Beispiel erklärt. Jedoch ist der Artikel sehr tiefgehend und nicht für einen oberflächlichen Überblick gedacht.
<u>Quelle:</u> http://www.pst.ifi.lmu.de/veroeffentlichungen/uwe.pdf

<u>Titel:</u> UML-based Web Engineering
<u>Autor:</u> Koch et. Al.LMU – Ludwig-Maximilians-Universität München
<u>Beschreibung:</u> Weiterführende Literaturliste und Projekte über UWE

Quelle: <http://uwe.pst.ifi.lmu.de/teachingCourseMaterial.html>

Titel: The Web Application Extension for UML

Autor: Paiano, R./Pandurino, A. (o. J.):

Beschreibung: Ein guter Überblick über WAE.

Quelle: <http://www.slideshare.net/dsbw2012/unit06wae2-13708972>

Allgemeine Literatur zu WebML

Titel: Model-Driven Web Engineering (WebML)

Autor: ETH

Beschreibung: Vorlesungsfolien von der ETH, welche einen guten Überblick über WebML geben

Quelle: <http://webeng.blog.globis.ethz.ch/files/2012/01/lec11-webml-print.pdf>

Titel: Modellierung mit WebML

Autor: Business Informatics Group TU Wien

Beschreibung: Vorlesungsfolien von der TU Wien, welche einen guten Überblick über WebML geben.

Quelle: http://big.tuwien.ac.at/teaching/offer/ss05/we_vo/weM11webML_print.pdf

Titel: Designing web applications with Webml and Webratio

Autor: Marco Brambilla, Sara Comai, Piero Fraternali, Maristella Matera

Beschreibung: Sehr detaillierte Beschreibung von WebML und die Umsetzung mit Web Ratio

Quelle: <http://www.webml.org/webml/upload/ent5/1/Chapter%209%20-%20WebML.pdf>

Titel: Web Modeling Language (WebML): a modeling language for designing Web sites

Autor: Stefano Ceri, Piero Fraternali, Aldo Bongio
Beschreibung: Beispielhafte Umsetzung von WebML, sehr technisch.
Quelle: http://www9.org/w9cdrom/177/177.html

Allgemeine Literatur zu OOWS

Titel: Developing Web Applications from Conceptual Models
Autor: Pelechano, V. et al. (o. J.)
Beschreibung: Ein kurzer, prägnanter Überblick über OOWS. Ist als Einstieg in das Thema geeignet.
Quelle: http://ceur-ws.org/Vol-74/files/FORUM_56.pdf

Titel: OOWS: A Method to Develop Web Applications from Web-Oriented Conceptual Models
Autor: Pastor, O./ Fons, J./ Pelechano, V. (o. J.)
Beschreibung: Führt den Leser anhand eines Fallbeispiels durch die Erstellung der Präsentations- und Navigationsmodelle.
Quelle: http://ceit.aut.ac.ir/~sa_hashemi/My%20Research/0-Selected%20Papers/2-ECommerce%20Systems/OOWS%20A%20Method%20to%20Develop%20Web%20Applications%20from%20Web-Oriented%20Conceptual%20_%206.pdf

Titel: OOWS 2.0: A Model-driven Web Engineering Method for the Development of Web 2.0 Applications
Autor: Francisco Valverde Giromé
Beschreibung: Präsentation über OOWS im Zusammenhang mit der Modellierung von Web-Anwendungen im Web 2.0
Quelle: http://www.slideshare.net/fravalgi/oows-20-a-model-driven-web-engineering-method-for-the-production-of-web-20-applications

Allgemeine Literatur zu OOHD

<u>Titel:</u> The Object-Oriented Hypermedia Design Model (OOHDM)
<u>Autor:</u> Daniel Schwabe, Gustavo Rossi
<u>Beschreibung:</u> Einstiegsdokument über OOHDM mit sehr vielen weiterführenden Links.
<u>Quelle:</u> http://oohdm.telemidia.puc-rio.br/

<u>Titel:</u> OOHDM
<u>Autor:</u> Nora Koch
<u>Beschreibung:</u> Skript von Nora Koch zur oberflächlichen Vorstellung von OOHDM.
<u>Quelle:</u> http://www.pst.ifi.lmu.de/lehre/SS98/se/koch/oohdm-folien.pdf

<u>Titel:</u> OOHDM Object Oriented Hypermedia Development Method
<u>Autor:</u> Frank Dopatka
<u>Beschreibung:</u> In dieser Arbeit geht es vor allem um die Beschreibung der einzelnen Phasen und deren Schematas und Notationen.
<u>Quelle:</u> http://www.frankdopatka.de/studium/koeln/oohdm.pdf

<u>Titel:</u> Model Driven Architecture with OOHDM
<u>Autor:</u> Hans Albrecht Schmid
<u>Beschreibung:</u> In dieser Ausarbeitung von Schmid geht es hauptsächlich um das Navigational Design und verschiedene Ansatzpunkte wie dieses umgesetzt werden kann.
<u>Quelle:</u> http://www.pst.informatik.uni-muenchen.de/~baumeist/icwe/ws/ws1/MDAwithOOHDM1405.pdf

<u>Titel:</u> Systematic Hypermedia Application Design with OOHDM
<u>Autor:</u> Daniel Schwabe, Gustavo Rossi, Simone Barbosa

Beschreibung: In dieser Ausarbeitung werden Hypermedia Anwendung in Bezug auf Design und Implementierung mit Fokus auf die Navigations- und Interface-Struktur analysiert. Schöne Gesamtübersicht mit vielen Referenzen.

Quelle:

https://www.google.de/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CDYQFjAB&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.38.7873%26rep%3Drep1%26type%3Dpdf&ei=l0fMUeL_Fc6rPNT2glg&usq=AFQjCNFWAy2SrfGEVGoKkL_ZQ3AkNdnj0Q&sig2=RdAzxt1E9sHcBJgydWVPTA

Titel: Hypermedia Design Methods

Autor: o.A.

Beschreibung: Eine gute zusammenfassende Präsentation zum Thema OOHDM mit allen wichtigen Informationen zum Thema für einen umfassenden Überblick.

Quelle:

<https://www.google.de/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CDQQFjAB&url=http%3A%2F%2Fwww.cs.man.ac.uk%2F~carole%2Fold%2Ffoohdm.ppt&ei=tU3MUaT4M8TsPJ2XgZgJ&usq=AFQjCNEDKuqzWLatbFOkNigQS-mJZy8uXQ&sig2=uWF6ThoKdoRppsOHXfL5xg>

Allgemeine Literatur zu den unbekannteren Web-Modellierungs Methoden

Titel: Model-driven web engineering methods: a literature review

Autor: Jesús Andrés Hincapié Londoño*, John Freddy Duitama

Beschreibung: Artikel über die bekannten Methoden und Vorgehensweisen allgemein und ab Seite 72 über WSDM und ab Seite 73 über OO-H

Quelle: http://www.scielo.org.co/scielo.php?pid=S0120-62302012000200007&script=sci_arttext&tlng=es

Titel: The OO-H Project

Beschreibung: Eine der wenigen Projektseiten über OO-H.

Autor: Web Engineering Group

Quelle: http://gplsi.dlsi.ua.es/iwad/ooh_project/index.htm

<u>Titel:</u> Towards Modeling of DataWeb Applications - A Requirements' Perspective
<u>Autor:</u> Werner Retschitzegger, Wieland Schwinger
<u>Beschreibung:</u> Überblick über die verschiedenen Methoden und Modelle. Eines der wenigen Papers, in dem HDM-Lite erwähnt und erklärt wird.
<u>Quelle:</u> http://users.dsic.upv.es/~west/iwmost01/files/contributions/UniversityLinz/AMCIS00.pdf

<u>Titel:</u> The Hera Research Program
<u>Autor:</u> The Hera Research Program
<u>Beschreibung:</u> Projektteam, welches sich mit Hera beschäftigt, allerdings wenige Informationen und längere Zeit nicht aktualisiert.
<u>Quelle:</u> http://wwwis.win.tue.nl/~hera/

<u>Titel:</u> Modeling User Input and Hypermedia Dynamics in Hera
<u>Autor:</u> Geert-Jan Houben, Flavius Frasinca, Peter Barna, and Richard Vdovjak
<u>Beschreibung:</u> Paper in dem die Hera Methode sehr detailliert und beispielhaft erklärt wird.
<u>Quelle:</u> http://people.few.eur.nl/frasinca/papers/ICWE2004a/icwe2004a.pdf

<u>Titel:</u> A Model-driven Approach for Designing Distributed Web Information Systems
<u>Autor:</u> Richard Vdovjak and Geert-Jan Houben
<u>Beschreibung:</u> Hera im Zusammenhang mit dem RDF.
<u>Quelle:</u> http://wise.vub.ac.be/sites/default/files/publications/icwe2005.pdf

<u>Titel:</u> An RMM-Based Methodology for Hypermedia Presentation Design
<u>Autor:</u> Flavius Frasinca, Geert Jan Houben, and Richard Vdovjak
<u>Beschreibung:</u> Sehr ausführliches Paper über RMM.
<u>Quelle:</u> http://wwwis.win.tue.nl/~houben/respub/adbis01.pdf

<u>Titel:</u> W2000: A Modeling Notation for Complex Web Applications
<u>Autor:</u> Luciano Baresi, Sebastiano Colazzo, Luca Mainetti, Sandro Morasca
<u>Beschreibung:</u> Beispielhaftes und ausführliches Paper über W2000.
<u>Quelle:</u> http://home.deib.polimi.it/baresi/papers/W2000Springer.pdf

<u>Titel:</u> An ontological approach to web application Design using w2000 methodology
<u>Autor:</u> Anna Lisa Guido, Roberto Paiano, Andrea Pandurino
<u>Beschreibung:</u> Beispielhaftes und ausführliches Paper über W2000.
<u>Quelle:</u> http://www.cs.ubbcluj.ro/~studia-i/2005-2/01-Paiano.pdf

<u>Titel:</u> From Design to Development: a W2000-based Framework; Issues and Guidelines
<u>Autor:</u> Paiano, R./Pandurino, A. (o. J.):
<u>Beschreibung:</u> Das Paper beschreibt inwiefern W2000 als Modellierungsmethode zur Implementierung von Webanwendungen genutzt werden kann. Dabei wird aufgezeigt wie die durch W2000 entwickelten Modelle erweitert und anschließend zur Kompilierung von Webanwendungen genutzt werden könnten. Die Entwicklung eines Frameworks, welches dies automatisiert ist von den Autoren angestrebt.
<u>Quelle:</u> http://dl.acm.org/citation.cfm?id=1038901

Literatur zur Generierung von Weboberflächen

<u>Titel:</u> Modellbasierte Generierung von Benutzungsoberflächen
<u>Autor:</u> Andreas Wolff (2011)
<u>Beschreibung:</u> Dissertation an der Universität Rostock, Besonderer Bezug auf Benutzungsoberflächen
<u>Quelle:</u> http://rosdok.uni-rostock.de/file/rosdok_derivate_000000004659/Dissertation_Wolff_2011.pdf

<u>Titel:</u> Extending UML to Model Navigation and Presentation in Web Applications
<u>Autor:</u> Nora Koch, Hubert Baumeister, Rolf Hennicker and Luis Mandel
<u>Beschreibung:</u> Detaillierung der UML Notationserweiterung für die Modellierung der Präsentationsebene von UWE.
<u>Quelle:</u> http://ufrsciencestech.u-bourgogne.fr/m2ecom/UWE/ExtendingUML.pdf ab Seite 2-5

<u>Titel:</u> Model-driven Development of Context-Aware Web Applications
<u>Autor:</u> Stefano Ceri, Florian Daniel, Maristella Matera, Federico M. Facca
<u>Beschreibung:</u> Paper über die Modellierung der Präsentationseben von WebML, insbesondere bei kontextabhängigen Web-Anwendungen.
<u>Quelle:</u> http://www.webml.org/webml/upload/ent5/2/TOIT-Matera-05-08.pdf

<u>Titel:</u> Developing GUIs with XUL
<u>Autor:</u> Eitan Suez
<u>Beschreibung:</u> Ein Foliensatz der einen kurzen Überblick über XUL gibt mit Beispiel und Referenzen auf die Mozilla Informationsseiten. Somit eine gute Zusammenfassung die offiziellen Quellen
<u>Quelle:</u> http://www.xmlaustin.org/wk/385/4/Eitan_Suez_XUL.pdf

<u>Titel:</u> XForms 1.0
<u>Autor:</u> W3C
<u>Beschreibung:</u> Originale Dokument von W3C Recommendations zum Thema
<u>Quelle:</u> http://www.re.be/css2xslfo/examples/xforms.pdf

<u>Titel:</u> XForms - The Next Generation of Internet Interfaces
<u>Autor:</u> Christoph Hofmann, Gregor Liebermann

Beschreibung: Eine Ausarbeitung zum Thema XFORMS was die offiziellen Dokumente vom W3C etwas oberflächlicher aber teilweise verständlicher Erklärt. Ein Gesamtüberblick.

Quelle: http://cm.hs-augsburg.de/report/2004/Hofmann_Christoph_Liebermann_Gregor/XForms.pdf

Titel: OOWS: A Method to Develop Web Applications from Web-Oriented Conceptual Models

Autor: Pastor, O./ Fons, J./ Pelechano, V. (o. J.)

Beschreibung: Führt den Leser anhand eines Fallbeispiels durch die Erstellung der Präsentations- und Navigationsmodelle.

Quelle: http://ceit.aut.ac.ir/~sa_hashemi/My%20Research/0-Selected%20Papers/2-ECommerce%20Systems/OOWS%20A%20Method%20to%20Develop%20Web%20Applications%20from%20Web-Oriented%20Conceptual%20_%206.pdf

7. Fazit

Abschließend und umfassend ist, nach der für eine Literaturrecherche gerechten Zusammenfassung mit einer Übersicht der gefundenen und verarbeiteten Literatur, das eigentliche Ziel der Arbeit noch einmal aufzugreifen und in einem Fazit zu rechtfertigen.

Das Ziel, des möglichst umfassenden Überblicks zu den aktuell bekannten theoretischen Ansätze und Methoden der modellbasierten Entwicklung von Webapplikationen, wurde im ersten Teil der Arbeit nach einer Einführung in das modellgestützte Web Engineering und der Modellierung von Web-Anwendungen durch eine Vorstellung der verschiedenen Methoden adressiert. In diesem Teil der Arbeit ist besonders deutlich geworden, dass es zwar seit Anfang der 90'er Jahre mehrere Ansätze der Neuentwicklung oder Weiterentwicklung von Modellierungs-Methoden bezogen auf Web-Anwendung gab, jedoch nur einige der Komplexität des Web-Engineering gerecht geworden sind und durch phasenunterteilte Methodik heute Anwendung finden. Diese ausgewählten Methoden und Ansätze sind, wie auch am Literaturumfang zu erkennen, bis heute die weitverbreitetsten Vorgehensweisen um der modernen Modellierung von Web-Anwendung eine Struktur zu geben und stehen deshalb auch im Fokus der Arbeit. Alle anderen Methoden, die in der Ausarbeitung erwähnt werden, sind zwar ebenso bekannt, sind aber in dieser Arbeit, neben dem geringeren Bekanntheitsgrad und weniger Literaturmaterial, vor allem durch fehlenden Schwerpunkt der Weboberflächen-Generierung von geringer Interesse.

Dieser genannte Schwerpunkt der Generierung von Weboberflächen beruht ebenso auf dem Ziel und Vorgaben der Arbeit. Daher wird hier ganz besonders noch einmal Fokus auf die bereits, durch die Literatur, stark hervorgehobenen Methoden gelegt und in Bezug auf die Ansätze zur Generierung von Weboberflächen untersucht. Die Untersuchungen bestätigten die bereits erarbeiteten Annahmen das vor allem UWE, WebML, OOHDM und OOWS durch ihre starke strukturelle Methodik der komplexen Anforderungen des Web Engineering gerecht werden.

Diese Ausarbeitung der Literaturrecherche zum Thema Web Engineering gibt somit durch ihren Fokus sehr deutlich an, dass UWE, WebML, OOWS sowie OOHDM besonders Beachtung in Verbindung mit modellgestützter Generierung von Web-Anwendungen mit Schwerpunkt Oberflächen verdient und belegt dies durch eine aussagekräftige Literaturübersicht für vertiefte Recherchen und Anwendung.

Literaturverzeichnis

- Alhir, S. S. (1998): UML in a Nutshell: A Desktop Quick Reference, Sebastopol: O'Reilly Media
- Davis, W. S./Yen, D. C. (2010): The Information System Consultant's Handbook: Systems Analysis and Design, Boca Raton: CRC Press
- Grüne, M. / Kneupner, R. (2002): Web Engineering in: WIRTSCHAFTSINFORMATIK 44 (2002) 3, S. 269–275
- Hitz, M. (2013): Eine Multikanal-Architektur für adaptive, webbasierte Frontendsysteme und deren Erweiterbarkeit durch Variantenbildung, DHBW Stuttgart
- Kappel, G. et.al. (2003): Web Engineering. Systematische Entwicklung von Webanwendungen, 1. Auflage, Heidelberg: DPunkt Verlag
- Kayser, S. (2007): XML Dokumenttyp-Definition kurz und bündig, München: Grin Verlag
- Koch et. al (2008): Web Engineering: Modeling and Implementing Web Applications, Chapter 7: UML-based Web Engineering – An Approach on Based Standards, London: Springer
- Paiano, R./Pandurino, A. (o. J.): From Design to Development: a W2000-based Framework; Issues and Guidelines, Lecce: Università di Lecce
- Pastor, O./ Fons, J./ Pelechano, V. (o. J.): OOWS: A Method to Develop Web Applications from Web-Oriented Conceptual Models, Department of Information Systems and Computation, Valencia: Technical University of Valencia
- Pelechano, V. et al. (o. J.): Developing Web Applications from Conceptual Models, Department of Information Systems and Computation, Valencia: Technical University of Valencia
- Pfeil, C. (2009): Adobe AIR: RIAs für den Desktop entwickeln Know-how für HTML/Ajax- und Flash, München: Pearson Deutschland GmbH
- Rossi, G. et.al. (2008): Web Engineering- Modeling and Implementing Web Applications, Human-Computer Interaction Series, London: Springer Verlag
- Vallecillo, A. /Koch, N. et.al. (2007): MDWEnet: A Practical Approach to Achieving Interoperability of Model-Driven Web Engineering

Methods,
<http://www.wit.at/people/schauerhuber/publications/mdwenet.pdf>, Abruf: 30.05.2013

Internetverzeichnis

- Brambilla, M. WebML: the Web Modeling Language - tutorial with audio and slides, <http://dbgroup.como.polimi.it/brambilla/webml>, Abruf: 01.06.2013
- Christoph Hofmann, C. / Liebermann, G. (2004) XForms - The Next Generation of Internet Interfaces, http://cm.hs-augsburg.de/report/2004/Hofmann_Christoph_Liebermann_Gregor/XForms.pdf, Abruf: 04.06.2013
- Frank Dopatka (2003): OOHD
Object Oriented Hypermedia
Development Method,
<http://www.frankdopatka.de/studium/koeln/oohtm.pdf>,
Abruf: 25.05.2013
- Heinrich, A. / Gooren-Sieber, S. (2011) Web Engineering
Aspekte der systematischen Entwicklung von
Web-Anwendungen, http://www.uni-bamberg.de/fileadmin/uni/fakultaeten/wiai_lehrstuehle/m Medieninformatik/Dateien/Sammelbaende/MI-Sem_WS2011-21_Web-Engineering.pdf, Abruf: 16.05.2013
- Koch, N. (1998) Object-Oriented Hypermedia Design
Method, <http://www.pst.ifi.lmu.de/lehre/SS98/se/koch/oohtm-folien.pdf>, Abruf: 20.06.2013
- Koch, N. (o.J): Classification of Modell Transformation Techniques used in UML-based Web Engineering,
http://delivery.acm.org/10.1145/1150000/1149997/a3-koch.pdf?ip=141.31.111.1&acc=ACTIVE%20SERVICE&key=C2716FEBFA981EF11136F45AFBA1AFF7E6AFF09410059491&CFID=343914638&CFTOKEN=51230859&acm__=1372428679_c1ac8dde4c22bdc63590a6f1b7bde5ff, Abruf: 20.05.2013
- Kraus, A./ Koch, N./ Knapp, A.(o. J.): Model-Driven Generation of Web Applications in UWE ,
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.66.9883>, Abruf: 25.05.2013
- Londoño, J./Duitama, J.(2012): Model-driven web engineering methods: a literature review,
http://www.sci.unal.edu.co/scielo.php?script=sci_arttext&pid=S0120-62302012000200007&lng=en&nrm=iso, Abruf: 01.06.2013
- Morovona, T. (2008): Diplomarbeit, Modellierung und Generierung von Web 2.0 Webanwendungen,

- <http://www.pst.ifi.lmu.de/~kochn/da/morozova.pdf>, Abruf: 25.05.2013
- o.A. Relationship Management Methodology (RMM): A Brief Introduction,
<http://www.cs.toronto.edu/~eharley/340F/RQ/GajenRMM.htm>, Abruf 17.0.2013
- Schlimbach, C. / Schölzel, S.(2008) UML Based Web Engineering (UWE), http://www.uni-trier.de/fileadmin/fb4/prof/INF/WI1/Lehrmaterialien/Web_Engineering/UWE.pdf, Abruf 18.05.2013
- Schmid, H.A. (o.J.) Model Driven Architecture with OOHDm,
<http://www.pst.informatik.uni-muenchen.de/~baumeist/icwe/ws/ws1/MDAwithOOHDm1405.pdf>, Abruf: 15.05.2013
- Schwabe, D./ Rossi, G./ Barbosa, S. (1996) Systematic Hypermedia Application Design with OOHDm,
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.38.7873> , Abruf: 14.06.2013
- Schwinger, W./Koch, N. (2003): Modeling Web Applications,
http://www.schwinger.at/PUBLICATIONS/04_2003_WebEngineeringBuch___Modellierung_von_Web_Anwendungen.pdf, Abruf: 10.06.2013
- Suez, E. (o.J.) Developing GUIs with XUL,
http://www.xmlaustin.org/wk/385/4/Eitan_Suez_XUL.pdf, Abruf: 04.06.2013
- Vallecillo, A. /Koch, N. et.al. (2007): MDWEnet: A Practical Approach to Achieving Interoperability of Model-Driven Web Engineering Methods,
<http://www.wit.at/people/schauerhuber/publications/mdwenet.pdf>, Abruf: 30.05.2013
- W3C (2003) XForms 1.0,
<http://www.re.be/css2xslfo/examples/xforms.pdf>, Abruf: 08.06.2013
- Wolff, A. (2010): Modellbasierte Generierung von Nutzeroberflächen,
http://www.swt.informatik.uni-rostock.de/deutsch/Mitarbeiter/andreas/diss_main.pdf, Abruf: 15.05.2013

A Comparison of SAP HANA with Relational Databases

Seminar Paper

Submitted on 07/02/2013

School of: Business
Program International Business Information Management
Course: WWI2010I

by
Simon Pieper und Can Paul Bineytioglu

BW Cooperative State University Stuttgart

Table of contents

List of abbreviations	3
Table of figures	4
1 Introduction	5
2 Introduction to databases	6
2.1 Introduction to relational databases	8
2.2 Assignment of permissions	9
2.3 Two-phase locking.....	12
2.4 Market Overview RDBMS.....	12
3 In-memory data management.....	16
3.1 Store data in main memory.....	17
3.2 Minimize data movement.....	18
3.3 Divide and conquer.....	23
4 SAP HANA	24
5 Comparison of SAP HANA with RDBMS	28
5.1 Hardware requirements	28
5.2 Management of data objects.....	30
5.3 Programming interfaces.....	31
5.4 Data persistence.....	32
5.5 Summary and additional aspects.....	34
6 Conclusion	35

List of abbreviations

2PL	2-Phase-Protocoll
CPU	Central Processing Unit
DB	Database
DBMS	Database Management System
DBS	Database System
I/O	Input/Output
MVCC	Multiversion Concurrency Control
OLAP	Online Analytical Processing
OLTP	Online-Transaction-Processing
RDBMS	Relational Database Management System
SQL	Structured Query Language
SSD	Solid State Disk

Table of figures

Figure 1 Components of a database	7
Figure 2 Database software market share in Germany in 2005 (in percentage)	13
Figure 3 RDBMS software revenues by vendor revenue estimates in 2006 (Millions of Dollars).....	14
Figure 4 Open Source database market share in 2012	15
Figure 5 Data access times of various storage types, relative to RAM.....	17
Figure 6 Illustration of dictionary compression.....	19
Figure 7 Row-oriented and column-oriented data access.....	22
Figure 8 Example of a query executed on a table in columnar storage.....	23
Figure 9 SAP HANA architecture.....	25
Figure 10 Save-points and logs in SAP HANA	33

1 Introduction

“Imagine you live in a major US city. Now, imagine that every time you want a glass of water, instead of getting it from the kitchen, you need to drive to the airport, get on a plane and fly to Germany, and pick up your water there. From the perspective of a modern CPU, accessing data which is in-memory is like getting water from the kitchen. Accessing a piece of data from the computer’s hard disk is like flying to Germany for your glass of water. In the past the prohibitive cost of main memory has made the flight to Germany necessary. The last few years, however, have seen a dramatic reduction in the cost per megabyte of main memory, finally making the glass of water in the kitchen a cost effective and much more convenient option.”¹

For many years enterprise applications have been used for transactional and analytical processes. Time by time relational databases were no longer able to perform as they have been doing the years before. The requests got more complex and the amount data (towards big data) grew. As a result research was done and several optimizations regarding Database Management Software and Hardware were done to improve the running systems. These Management systems traditionally organized the data row-oriented but were developed to run column-organized as well. “On the other hand significantly more main memory is available to applications that allow to store the complete compressed database of an enterprise in combination with the equally significantly enhanced performance.”² Now, another approach, the in-memory database, appeared on the market and was able to process these huge analytical queries in less than a second. In comparison to the relational approach it was able to capture new business processes and applications. Reasoned the question appears whether the strong differentiation between OLAP and OLTP is still useful or the requests are managed on connected records.

Moreover, in-memory technology promises many benefits for the use in enterprise applications. However, it is controversially discussed as there are also not many customer implementations to refer to.

In this paper we will briefly introduce databases and the basic characteristics about relational data models. As the in-memory approach gets more and more interesting according to the development of cloud computing and is much faster than the conventional relational database in terms of data processing, the second part of this paper will discuss fundamentals and advantages of the in-memory concept. Furthermore both approaches will be compared and summarized in a final table.³

¹ Plattner, H./Zeier, A. (2012), p. xix

² Krueger, J. et. al. (2011), p. 547

³ Cf. Krueger, J. et. al. (2011), p. 547

2 Introduction to databases

To get a deeper understanding of the following topic and comparison of in-memory databases to relational databases, a brief explanation of (relational) databases will be given. Main ideas and issues will be discussed from the big picture perspective, shown with practical business examples. For further information the given sources can be prospected.

Why do we need databases?

The main reasons for the development of database systems are:⁴

1. The amount of data rises. Databases keep data compromised.
2. The structure of the data gets more and more complex.
3. Actual/Relevant data needs to be updated frequently and as fast as possible.
4. Data should be managed independently on the application.

The term database is used as a synonym for a database system (DBS). It consists of the database itself (data basis) and a Database Management Software (DBMS). A database system is a conglomerate of data, which can be accessed through all users or applications.⁵ It is an electronic system for administrating (in most cases) a large amount of data. It should fulfill the following requirements:⁶

1. Handle big data efficiently, consistently and permanently
2. Summarize partial quantities in databases
3. Project logical relations digitally
4. Customize user and application permissions

Standardized rules explain how to store data. Different types of files can be stored – from simple text files to pictures, audio files and videos. A database system consists of the two parts:^{7 8}

1. A data basis: All provided information is bundled in data sets, called records, and stored in the data basis.

⁴ Cf. w. a. (w. y.b)

⁵ Cf. Leve, J. (2007)

⁶ Cf. w. a. (w. y.a)

⁷ Cf. Elmasri, R./Navathe, S. (2007)

⁸ Cf. w. a. (w. y.a)

2. A Database Management System (DBMS): The software installed for the database, which is responsible for managing the database.

A data basis describes the data of a database, a physical amount of files where the data and the data dictionary are saved. Those should be stored in the file system.

A data dictionary consists of meta data describing the data basis (e.g. as a set of tables). Additionally it contains data about the use and meaning of the applied data model, the data relations and integrity.

A Database Management System is software for managing the data basis. All processes are managed and different services are provided. Through implementing parameters the system can be customized on the users and applications needs. These can be functions for storing the data, managing the resources and guaranteeing data consistency. Important questions are:

- How many users work with the database?
- What kind of data and how much information must be stored?
- What kind of security settings is needed? How should it be configured?

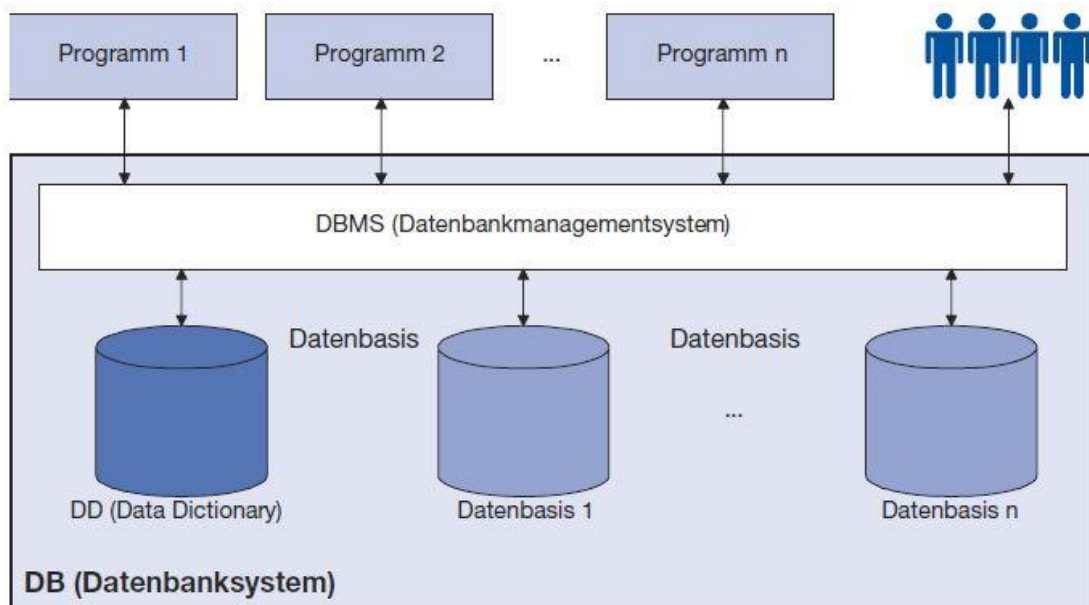


Figure 1 Components of a database⁹

⁹ Cf. Elmasri, R./Navathe, S. (2007)

2.1 Introduction to relational databases

As set in the assignment, the following paper will handle the second type of databases, the relational database. A relational database is a database using the relational data model.

A “relational data model is a data abstraction that presents the data in a database as a set of tables, which are also called relations. Each table has a name and contains a special top row and a finite number of data rows [which are called tuples and are the data sets]. Each entry of the top row of a table is called an attribute [and projected as a column]. The number of attributes of a relation is called its arity or dimension [...]”¹⁰

Connections (relations) between the tables are done within using primary keys. A deeper look at the relational model is not necessary for the comparison with in-memory databases. The Data model is the theoretical fundament for a database and describes in which way data is stored and arranged in the database. Today a data model can be very complex, dependent on the amount of information and how they must be processed. It consists of the following three aspects:¹¹

1. Generic data structure
2. Generic operators
3. Integrity conditions

Common other database models are:¹²

1. Object oriented model
2. Hierarchical database model
3. Network model
4. Document model

There are several mixtures of those concepts, e.g. object-relational. The most common data model is the relational data model, which is focused in this paper.

Traditionally relational databases work **row-oriented**. If the user starts a query, the data is read from row to row. In Case of OLTP reading full rows is a benefit.¹³ This results in a “full column read”, every single column is automatically read as well. The larger the amount of data is, the longer it takes to read every column.

¹⁰ Revesz, P. (2010), p. 15

¹¹ Cf. w. a. (w, y,b)

¹² Cf. w. a. (2010)

¹³ Cf. w. a. (2013a)

Instead of working row-oriented some relational databases can work **column-oriented**. Data is written in columns, which simplifies the queries. Now, only columns are read that are explicitly mentioned in the query. Furthermore data is better compromised and often less I/O is generated through using this approach.¹⁴

Database architecture

The database architecture depends on how a database system must be configured, for a single user or for a cluster of users (a network). It is essential to choose the right architecture, it decides on the quality and quantity of the services and security issues. Furthermore it is necessary for an appropriate database design and the costs. In general it could be said, that the “better” and more customized the architecture is, the more it will cost.¹⁵

Database design

A requirement analysis informs about the performance a database should deliver now and in the future. Main issue for the database developer is to collect and analyze all information and to put in context.

A useful database design is characterized through a data model, which is easy to maintain and to extend for the database developer. Possible questions are:¹⁶

- Which are the objects and attributes we need?
- How do the relations between these objects look like?
- How many objects do we need?
- What are the typical operations done for this system?

There is no recipe available for each database design, every project is unique and needs to fulfill its own requirements.

2.2 Assignment of permissions

The assignment of permissions is one the most important configuration items before implementing databases. The decision between a single-user and multi-user approach depends on the amount of users, permissions and the amount of data stored in the database.

¹⁴ Cf. Rohmoser, K. (w.y.)

¹⁵ Cf. w. a. (w. y.b)

¹⁶ Cf. w. a. (w. y.a)

Stand-alone system

Stand-alone systems are local and can be accessed by only one client/program at once. A network access is in theory possible, but is not recommended as only one access at the same time can be processed. Advantage of a stand-alone system is that no time-consuming configuration for a multi-user approach and no transaction management (needed if more than one operation is done through many users on the data basis at once) is necessary. An implementation is cheaper than a multi-user approach.

A stand-alone system is suitable for databases with a small amount of data and only single-user operations. A server based system is not necessary as the resources are not needed and the system would be too expensive.^{17 18}

Multi-User system

A multi-user system is much more complex than a single-user approach and can be very expensive. Regarding access speed and data security, specific configurations need to be done. The multi-user approach can be classified into two different systems:^{19 20}

a) File-Server database

A File-Server database is also known as a File-Share system. The database is located as an accessible file on a server, which can be reached through the network. The whole database can be queried and sent to the client. Further operations as filtering the required data are done on the client's system. Therefore the server is a distributor sharing the original database file with the connected computers. Disadvantage is that the network load is extremely high as not only the needed data is sent to the client. As a first step the user will get the complete data inventory. If the user only reads data, the problem can be handled by implementing a useful network infrastructure. If he changes the data and sends the complete data back to the server, it will be difficult to check the integrity of the data base or refusing other queries or changes. Microsoft Office Access and Open Office Base are the most common File-Share Systems.

b) Client-Server database

The data base server is a central computer containing both the data and the Data Base Management System (DBMS). Every connected client/program can access this computer through the network. The main difference to the File-Share System is that the client only re-

¹⁷ Cf. Webb, M. (w. y.)

¹⁸ Cf. w. a. (2009)

¹⁹ Cf. w. a. (2009)

²⁰ Cf. Webb, M. (w. y.)

quests the data he needs and not the complete amount of data available. The operations are done on the central server, accesses are done indirectly as the DBMS server is between the data base and the clients. The TCP/IP Protocol is typically used nowadays.

The advantages are:

- A lower network load as there are only SQL commands sent to the DBMS server.
- The possibility to immediately check the data integrity. Changes can be checked as the client sends his transactions to the DBMS where they are processed.
- The data base server is scalable and allows higher traffic.
- Less hardware requirements for the clients as only results are sent to the user.

A client-server model is suitable if many clients and request are done. Otherwise the high costs for server and software are not reasonable.

If applying a multi-user system three different, common approaches are available:^{21 22 23}

Optimistic Locking

Every user can do every action, the table is open inside the network for every session. When a transaction accesses the server the data base resource is not locked for every other transaction. Within considering the ROWIDS the DBMS has to decide which user can access what kind of data. This approach can be very productive, if the concept is developed carefully and guarantees the records are correctly edited.

Pessimistic Locking

Every user can still do every action, but the DBMS locks the database/record as soon as the first transaction is started. The table is open inside the network for only the current session. "The other session users can not edit [at] the same."²⁴ When a transaction accesses the server, the database resource is locked for every other transaction as long as the first transaction is not finished.²⁵ Disadvantage of this „secure“ solution is, that the productivity of the data base decreases.

Release Lock as another type of locking is rather uncommon. Every user can do every action. The developer has to implement the permissions in the source code.

²¹ Cf. w. a. (w. y.c)

²² Cf. w. a. (w. y.e)

²³ Cf. w. a. (w. y.f)

²⁴ w. a. (w. y.c)

²⁵ Cf. w. a. (2006)

2.3 Two-phase locking

Two-phase locking (2PL) is a type of a concurrency protocol which is often used for database systems. “Locking is a mechanism commonly used to solve the problem of synchronizing access to shared data.”²⁶ Transactions are divided into two phases:^{27 28}

1. “Grow phase”: The transaction requests a lock.
2. “Decrease phase”: The transaction releases a lock.

This means that if a transaction starts, a lock is acquired and kept until the transaction is done. The scheduler will not give another transaction a lock until the first one released the lock. Therefore it is ensured that only one transaction keeps a lock at once, a data item can only be changed by one single transaction at time. Serializability is guaranteed.²⁹ Three different type of locks exist:

1. Read lock (SLOCK³⁰)
2. Write lock (XLOCK³¹)
3. UNLOCK

After a transaction released a lock it cannot request another lock. SLOCK and XLOCK are both part of the “grow phase” and are done before the “decrease phase” is started. A DEAD-LOCK may be a problem of a 2-phase-protocoll.

2.4 Market Overview RDBMS

The German market

Judging about market share in the RDBMS software market depends “upon who you ask and how you phrase the question.”³² The long-term market leader Oracle is “offended” from Microsoft and IBM. Most of the reports say Oracle is still leading the RDBMS database segment.

²⁶ w. a. (w. y.e)

²⁷ Cf. w.a. (2010)

²⁸ Cf. w.a. (w. y.f)

²⁹ Cf. w.a. (2010)

³⁰ An object can be read by many transactions and as a result be locked by several SLOCKS. But once a SLOCK is set a XLOCK is not possible (cf. FH Köln, (2010)).

³¹ A XLOCK is set if a transaction starts a “writing access”. An object can be edited by only one transaction per time. No other transaction can set another lock on this object until the first lock is released (cf. FH Köln, (2010)).

³² <http://databases.about.com/library/weekly/aa060401a.htm>

A brief look at the German DBMS market in 2005 shows that Oracle and IBM had with 33% the same market share. Microsoft had a market share of 19%. Therefore the “big three” totaled a market share of 85%.

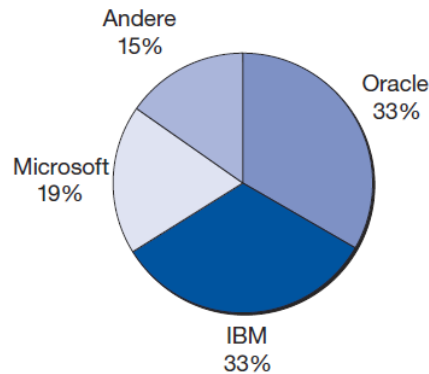


Abbildung 1.17: Datenbankmarkt Deutschland 2005 (Quelle: Gartner)

Figure 2 Database software market share in Germany in 2005 (in percentage) ^{33 34}

Worldwide market

In 2006 the total software revenue added up to 15,21 Billion US-Dollars and was dominated by the three vendors Oracle, IBM and Microsoft with an overall market share of 85,6%. Oracle had with a revenue of 7,17 Billion Dollars a market share of 47,1% and was by far the market leader in a worldwide consideration. IBM, number 2 worldwide in this segment in 2006, had with a revenue of 3,20 Billion Dollars a market share of 21,1%. Third, Microsoft had with a revenue of 2,65 Billion Dollars a market share of 17,4%. Teradata and Sybase had both a market share of 3,2% with a revenue of about 490 Million Dollars.

³³ Cf. Elmasri, R./Navathe, S. (2007), p. 48

³⁴ Newer data for the German DBMS market could not be found.

Company	2006 Market Share (%)		2005 Market Share (%)		2005-2006 Growth (%)
	2006	2005	2006	2005	
Oracle	7,168.0	47.1	6,238.2	46.8	14.9
IBM	3,204.1	21.1	2,945.7	22.1	8.8
Microsoft	2,654.4	17.4	2,073.2	15.6	28.0
Teradata	494.2	3.2	467.6	3.5	5.7
Sybase	486.7	3.2	449.9	3.4	8.2
Other Vendors	1,206.3	7.9	1,149.0	8.6	5.0
Total	15,213.7	100.0	13,323.5	100.0	14.2

Figure 3 RDBMS software revenues by vendor revenue estimates in 2006 (Millions of Dollars)³⁵

In 2012 the worldwide market grew and changed in comparison to 2006. Oracle's market share decreased to 33,8% and Microsoft minimized the distance to Oracle with a market share of 30%. Only Microsoft's SQL Server product had a market share of 13,9% as it is only available on Windows running systems. Microsoft focuses on its own systems and tries to leave the other platforms to its competitors. In the next years it is predicted that Microsoft will gain more market share for the Windows running systems. "Their SQL Server product becomes more reliable year after year and is especially appealing to Microsoft network administrators who appreciate the product's familiar brand name and the distinctive Microsoft "look and feel"."³⁶

After focusing on the market situations of the commercial DBMS software and the development in the last years, a brief look at the market shares for open source DBMS software is appropriate. MySQL by Oracle (also available as a commercial enterprise DBMS software) has a market share of 56% and clearly dominates the market. MariaDB (as a separation of MySQL) by Monty Program AB has a market share of 15%. MongoDB (document-oriented, most popular NoSQL database) by 10gen has a market share of 16%. PostgreSQL (object-relational) by PostgreSQL Global Development Group (open source community) has a market share of 10%. CouchDB (document-oriented) by the Apache Software Foundation has the lowest market share with 3%.

³⁵ Cf. Pettey, C. (2007)

³⁶ w. a. (2013b)

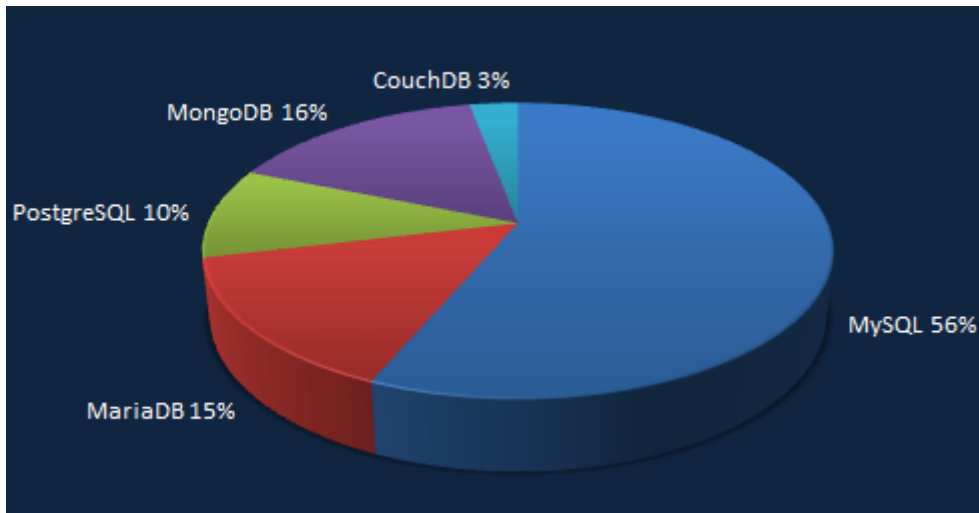


Figure 4 Open Source database market share in 2012³⁷

³⁷ Cf. w.a. (2012)

3 In-memory data management

Storing data in main memory³⁸ is not a new concept. What is new about in-memory computing is that the whole operational or analytic database can be stored in main memory (RAM) as the primary persistence layer.³⁹

In-memory computing is a technology that allows the processing of huge amounts of data in main memory in order to obtain immediate results from analysis and transaction. The data to be processed is ideally real-time data – that is, “data that is available for processing or analysis immediately after it is created.”⁴⁰

In-memory technology aims at uniting operational processing and analytical processing in one database management system for enterprise applications.⁴¹

The in-memory revolution is not simply about putting data into main memory and thereby being able to process data faster. The in-memory revolution basically emerged from two major developments in the IT industry in the last decades:⁴²

1. The emergence of multi-core central processing units (CPU) and the necessity of leveraging this parallelization in data processing;
2. Cost of main memory has dropped significantly and capacity of main memory nowadays reaches disk size with up to several terabytes.

In order to leverage potential performance improvements, the following three basic concepts are followed in in-memory computing:⁴³

- Keep data in main memory (Random Access Memory, RAM) to speed up data access.
- Minimize data movement by leveraging the columnar data storage concept, compression, and performing calculations at the database level.
- Divide and conquer: Leverage the multi-core architecture of modern processors and multi-processor servers and even scale out into a distributed landscape.

³⁸ “Main memory refers to silicon-based storage directly accessible from the CPU while in-memory refers to the concept of storing the primary data copy of a database in main memory. Main memory is volatile as data is lost upon power failure.”, Plattner, H./Zeier, A. (2012), p. 6

³⁹ Cf. Word, J. (2012), p. 26

⁴⁰ Cf. Vey, G./Krutov, I. (2012), p. 7

⁴¹ Cf. Plattner, H./Zeier, A. (2012), p. 29

⁴² Cf. Plattner, H./Zeier, A. (2012), p. xix

⁴³ Cf. Vey, G./Krutov, I. (2012), p. 7

3.1 Store data in main memory

Over the last few decades the capacity of main memory in servers has continuously increased while the prices have significantly decreased at the same time.⁴⁴ Nowadays servers have main memories that can reach hard disk capacities.⁴⁵ As capacities of 2 terabytes of main memory can nowadays be considered as standard, and single enterprise class servers can hold even several terabytes of main memory, databases can entirely be moved into main memory.

Requests can be processed directly from main memory without using any hard disk. Such databases are referred to as In-Memory-Databases (IMDB) or Main-Memory-Databases (MMDB).⁴⁶

Using main memory as the data storage location

One major reason to use the main memory for data storage and management is that accessing data stored in main memory is much faster than accessing data stored on disk.⁴⁷ The following figure 5 shows the access times for data stored in different locations.

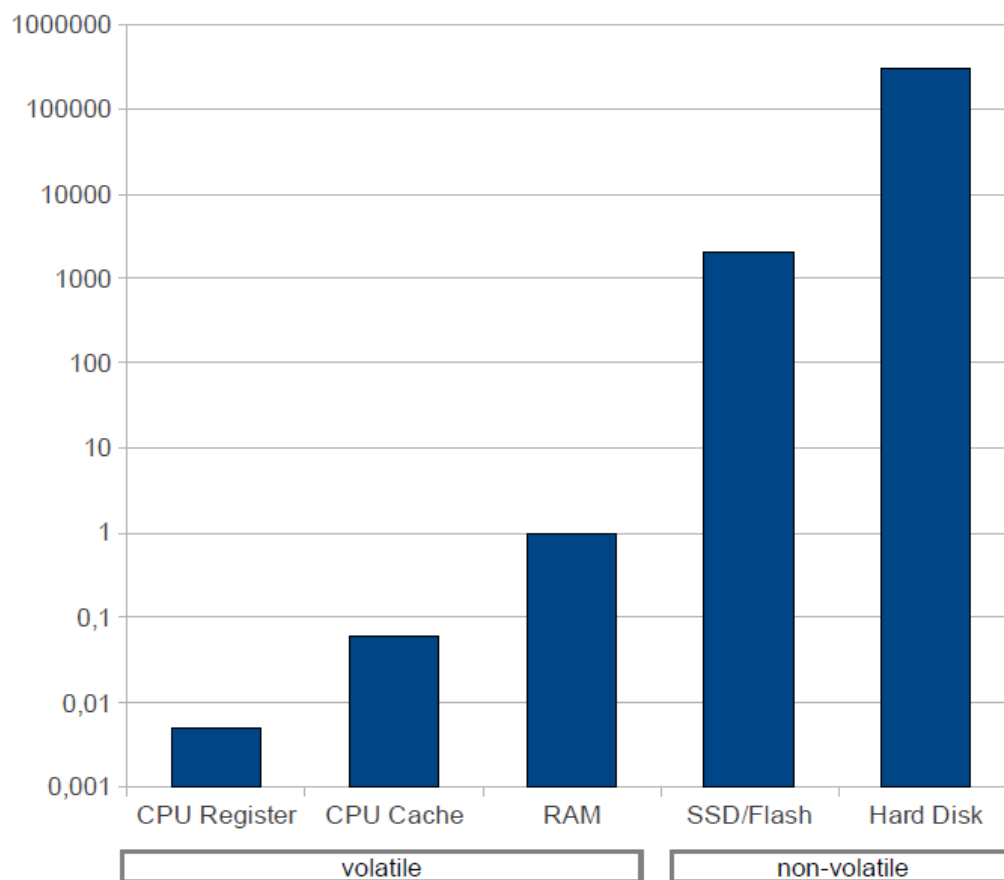


Figure 5 Data access times of various storage types, relative to RAM⁴⁸

⁴⁴ Cf. SAP (2012), p. 6

⁴⁵ Cf. Krueger, J. et al. (2011), p. 548

⁴⁶ Cf. ibidem, p. 548

⁴⁷ Cf. Vey, G./Krutov, I. (2012), p. 8

As indicated in figure 5, the main memory (RAM) is the fastest storage location that can hold a large amount of data. Although CPU registers and CPU caches may access and process data faster, their usage is limited to the actual processing data as opposed to RAM. Data in main memory can be accessed more than a hundred thousand times faster than data on a hard disk. A high-speed bus connects the main memory directly to the CPUs, whereas hard disks are connected via a chain of buses and controllers.⁴⁹

Thus, in terms of access time keeping the data in main memory improves the database performance as opposed to keeping data on disk.

Persistence Layer

The SAP HANA database persistence layer stores data in persistent disk volumes, which can be either hard disk or solid-state drives (SSD). The persistence layer ensures that changes are durable and that the database can be restored to the most recent committed state after a restart. An advanced delta-insert approach is used for rapid backup and logging. If power is lost, the data in RAM is lost. However, because the persistence layer manages save-points and backup at such high speeds (from RAM to SSD) and recovery from disk to RAM is so much faster than from regular disk, managing data in main memory does not impose any severe difficulty in terms of data persistence.⁵⁰

Section 5.4 will outline how data persistence is accomplished with SAP HANA.

3.2 Minimize data movement

Compression

Data compression methods leverage redundancy within data. Compression techniques can be particularly well applied to columnar storage⁵¹ in an enterprise data management scenario due to two reasons:

1. All data within a column have the same data type;
2. Often there are only few distinct values, e. g. in a country or status column.

The two main reasons for using compression techniques are saving storage space and increasing database performance. As the cost of main memory is still expensive in comparison to disk, efficient use of space is important. Through compression, more information can be loaded for processing at the same time and thus performance can be increased.⁵² The goal of the data compression is to minimize data movement from RAM to the processor.⁵³

⁴⁸ Contained in: Vey, G./Krutov, I. (2012), p. 8

⁴⁹ Cf. Vey, G./Krutov, I. (2012), p. 8

⁵⁰ Cf. Word, J. (2012), p. 28

⁵¹ Columnar data storage is explained later in this paper.

⁵² Cf. Plattner, H./Zeier, A. (2012), p. xx

⁵³ Cf. Vey, G./Krutov, I. (2012), p. 10

Dictionaries are used to represent text as integer numbers. The database can thereby compress data, while not imposing additional CPU load for decompression. This is illustrated in figure 6.

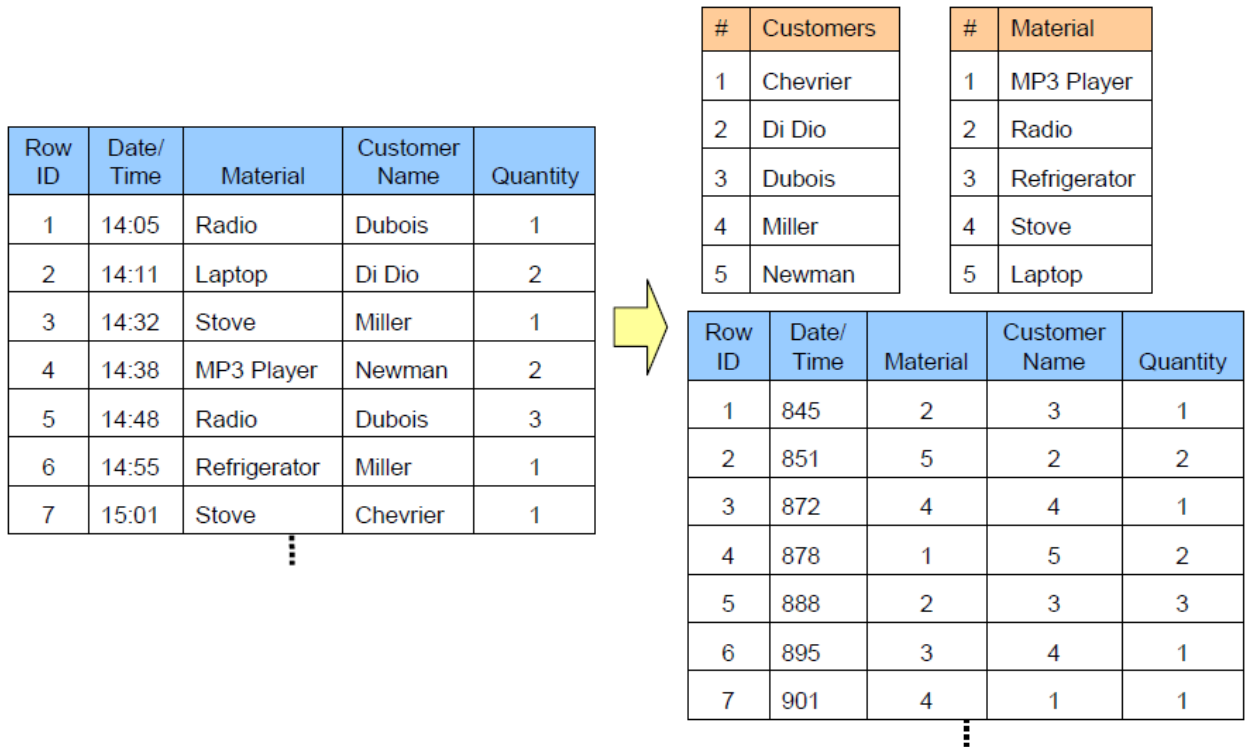


Figure 6 Illustration of dictionary compression⁵⁴

The table on the left-hand side of the figure contains the original representation of the data with text attributes. The two small tables on the upper right side are dictionaries that assign an integer value to each distinct attribute value. In the table on the lower right side, the text values are replaced by the corresponding integer values according to the dictionary. Using dictionaries for (not only) text attributes reduces the table size as each distinct attribute value needs only to be stored once in the dictionary. All additional occurrences refer to the corresponding integer value.⁵⁵

While attributes with few distinct values are compressed well, attributes with many distinct values do not increase the performance much when being compressed.

Diverse compression techniques, also more effective ones, may be employed with in-memory computing. However, in all compression methods, the balance between compression effectiveness (which results in more availability of storage capacity), the degree of data movement (which results in higher performance), the amount of resources required for decompression, as well as data accessibility need to be considered in relation.

⁵⁴ Contained in: Vey, G./Krutov, I. (2012), p. 10

⁵⁵ Cf. Vey, G./Krutov, I. (2012), p. 10

Columnar and row-based storage

In databases tables are stored row-based or columnar. A row-oriented storage system organizes a table as a sequence of rows. In contrast, columnar storage stores the data in a sequence of columns.⁵⁶

Due to the growing demand for analytical functionalities such as ad-hoc requests for transactional data, column-oriented database systems have become more and more important in recent years.⁵⁷

A main advantage in columnar storage is that only relevant columns need to be read as opposed to requests in row-oriented structures on hard drives where all attributes of a row, although irrelevant, are read.⁵⁸

Data can be accessed faster simply through reduction of the data volume that is read.

The following table shows benefits and drawbacks of the row-based and column-based storage concepts.

	Row-based storage	Columnar storage
Advantages	<ul style="list-style-type: none"> Record data is stored together Easy to insert/update 	<ul style="list-style-type: none"> Only affected columns have to be read during the selection process of a query Efficient projections⁵⁹ Any column can serve as an index
Drawbacks	<ul style="list-style-type: none"> All data (i. e. also irrelevant data) has to be read during execution of query 	<ul style="list-style-type: none"> After selection, selected rows have to be reconstructed from columns No easy insert/update

Table 1: Advantages and drawbacks of row-based and columnar storage⁶⁰

Row-based data storage is recommended if.⁶¹

- The table has a small number of rows, such as configuration tables.

⁵⁶ Cf. Vey, G./Krutov, I. (2012), p. 11; Cf. Word, J. (2012), p. 27

⁵⁷ Cf. Krueger, J. et al. (2011), p. 549

⁵⁸ Cf. Krueger, J. et al. (2011), p. 549

⁵⁹ A projection is a view on a table with a specific subset of columns

⁶⁰ With modifications taken from: Vey, G./Krutov, I. (2012), p. 12

⁶¹ Word, J. (2012), p. 27 f.; Cf. SAP (2012), p. 13 f.

- The application processes only a single row at a time (several selects or updates of single records).
- The application accesses the complete record.
- The compression rate would be low, which is the case when the columns contain mainly distinct values.
- Aggregations and fast searching are not required.

Row-based data storage is used, for example, for SAP HANA database metadata and for configuration data.

Columnar data storage is recommended if:⁶²

- Calculations are performed on a single column or a few columns only.
- Queries are based on the values of a few columns.
- The table has a large number of columns.
- The table has a large number of rows, and column-based operations are executed (e. g. aggregate, scan).
- The majority of columns contain only a few distinct values (compared to the number of rows), resulting in higher compression rates.

The following figure illustrates row-based and column-based storage.

⁶² Cf. Word, J. (2012), p. 27 f.; Cf. SAP (2012), p. 13 f.

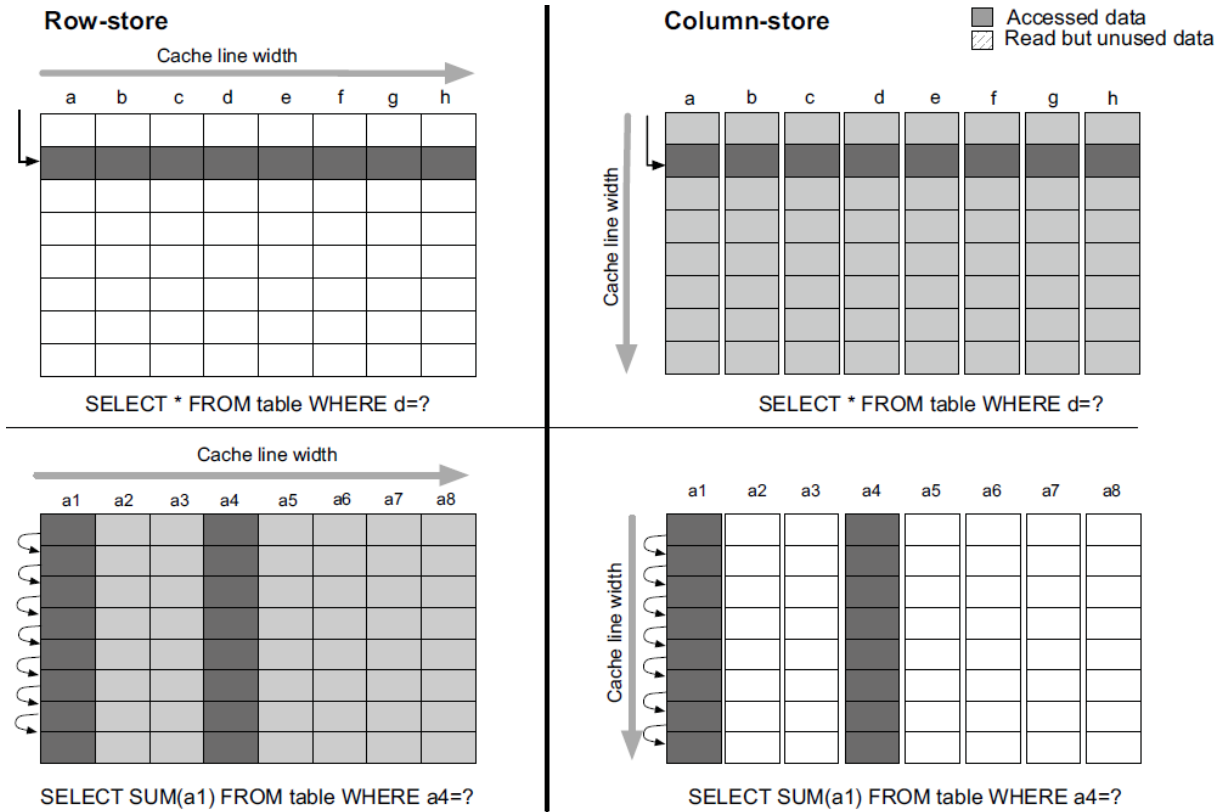


Figure 7 Row-oriented and column-oriented data access⁶³

The following figure shows an example of a query executed on a table in columnar storage.

⁶³ Contained in: Krueger, J. et al. (2011), p. 550

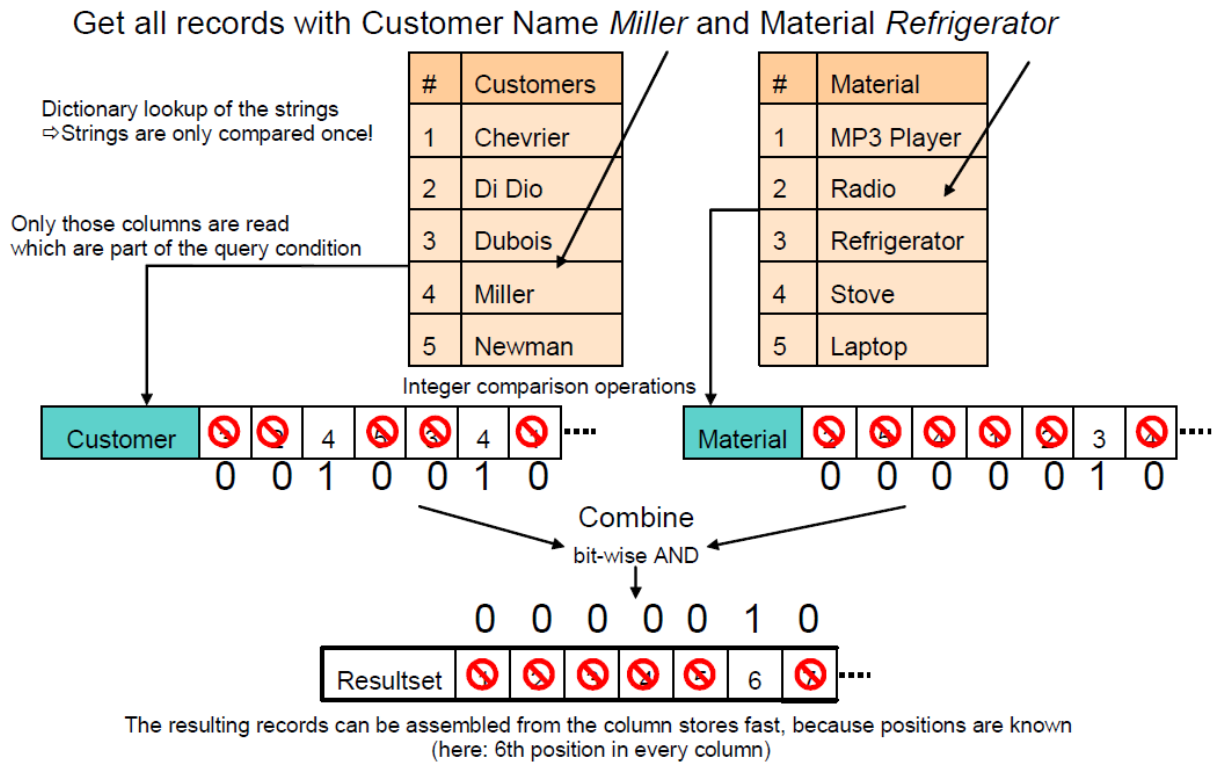


Figure 8 Example of a query executed on a table in columnar storage⁶⁴

3.3 Divide and conquer

Parallelization across multiple cores

As CPU cores are no longer becoming any faster, but the number of CPU cores is increasing over time, parallel processing capabilities of multi-core CPUs are of particular importance to the development of data processing and in general to future software development.⁶⁵

Rows of a table in a relational database can be processed independently from each other, which allows parallel processing. When scanning a table for specific attribute values in the form of a query, the table or the set of columns relevant to the query can be divided into subsets. The processing of the query can be parallelized by spreading the subsets across multiple cores and machines. Thus, queries are processed faster. Also, the data that is accessed can reside on diverse machines.

⁶⁴ Contained in: Vey, G./Krutov, I. (2012), p. 12

⁶⁵ Cf. Plattner, H./Zeier, A. (2012), p. xx

4 SAP HANA

This section will introduce SAP HANA and its key characteristics with regards to the comparison with relational DBMS in the fifth chapter. In the following the terms SAP HANA Appliance and SAP HANA database are defined to ensure a better understanding of this chapter:

- *SAP HANA Appliance:*

“SAP HANA is a flexible, data source agnostic appliance that allows you to analyze large volumes of data in real time, without the need to materialize aggregations. It is a combination of hardware and software, and it is delivered as an optimized appliance in cooperation with SAP’s hardware partners for SAP HANA.”⁶⁶

- *SAP HANA database, also referred to as the SAP in-memory database:*

“The SAP in-memory database is a hybrid in-memory database that combines row-based, column-based, and object-based database technology, optimized to exploit the parallel processing capabilities of current hardware. It is the heart of SAP offerings like SAP HANA.”⁶⁷

The SAP HANA database can be considered as the core of the SAP HANA Appliance to support complex business analytical processes in combination with transactionally consistent operational workloads.

From a technical perspective, the SAP HANA database consists of multiple engines for data processing that use a distributed query processing environment.

The following figure gives an overview of the SAP HANA architecture.

⁶⁶ Vey, G./Krutov, I. (2012), p. 16

⁶⁷ Ibidem, p. 16

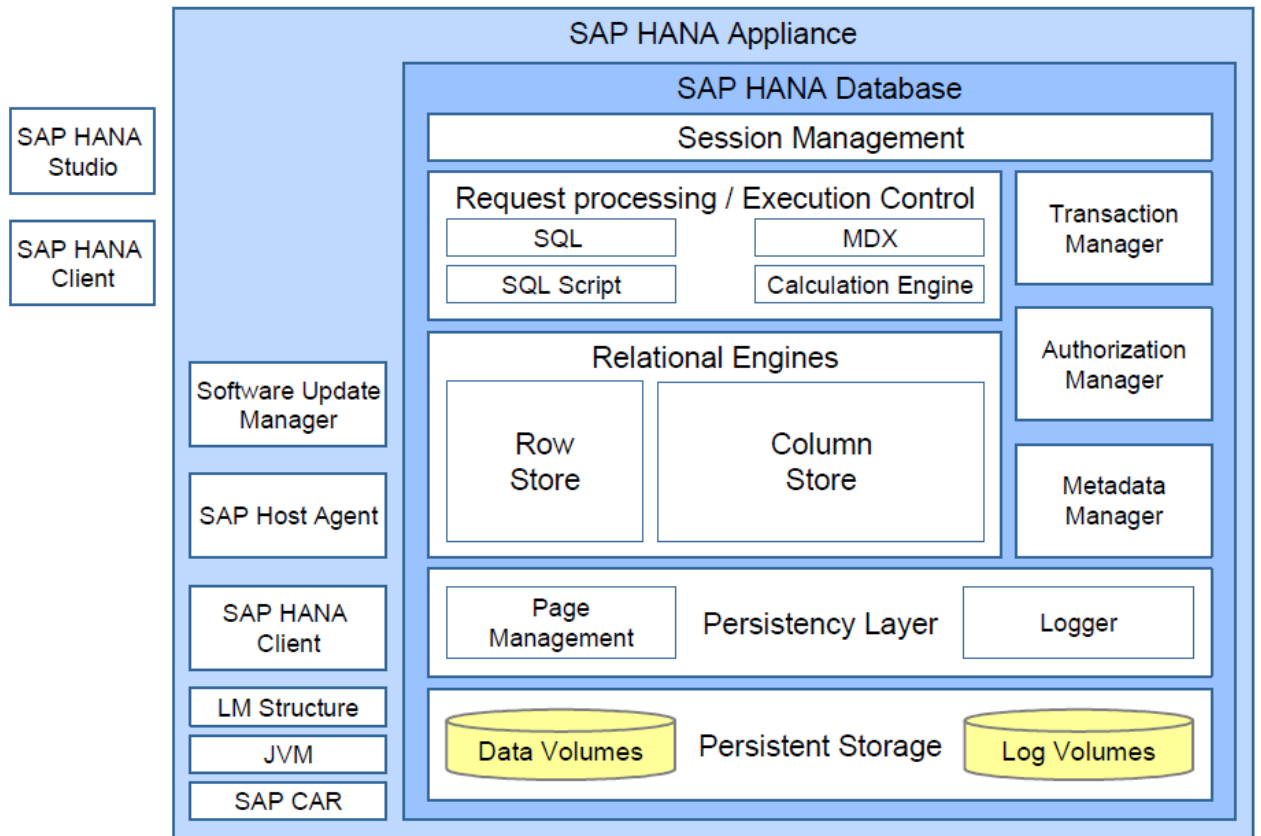


Figure 9 SAP HANA architecture⁶⁸

SAP HANA database

SAP HANA has both columnar and row-based data storage within the same engine, which is one differentiating attribute in comparison to traditional databases.⁶⁹

With SAP HANA you can choose between column and row store – depending on what is more appropriate for your use case (see also advantages&drawbacks in section X)^{70 71}

The appropriate engine for data storage can be chosen for every table at the time of creation of a table. Row-based stored tables are loaded at start-up whereas columnar stored tables can be loaded at start-up or during normal operation of the SAP HANA database.

A common persistency layer underlies the two engines involving page management and a logging system. Modifications to the database tables are written to the data volumes on persistent storage through savepoints.

⁶⁸ Contained in: Vey, G./Krutov, I. (2012), p. 16

⁶⁹ Cf. SAP (2012), p. 13

⁷⁰ Cf. ibidem, p. 13; Cf. Word, J. (2012), p. 27

⁷¹ Cf. Word, J. (2012), p. 27

Moreover, for every committed transaction a log entry is persisted to the log volumes on the persistent storage. Flash technology storage is used for the log volumes for high I/O performance and low latency.

To access the relational engines the SAP HANA database supports SQL (JDBC/ODBC), MDX (ODBO), and BICS (SQL DBC). Through the calculation engine calculations can be performed in the database without moving the data into the application layer.

SAP HANA appliance

The SAP HANA appliance consists of diverse components. A major component is the database that is described above. Additional components are required to manage and operate the database – containing e. g. installation files for the SAP HANA Studio⁷² and several libraries used to connect clients and applications to the SAP HANA database.⁷³

Following client libraries are used to access the SAP HANA database via SQL or MDX:⁷⁴

- JDBC driver for Java clients
- ODBC driver for Windows/Unix/Linux clients, especially for the integration of Microsoft Office applications
- DSBL (Database Shared Library) for ABAP

The following section outlines some core distinctive features of the SAP HANA database to show the key differentiators to common relational database management systems.

- *Multi-engine query processing environment:*
SAP HANA provides a multi-engine query processing environment in order to fulfil the requirements of managing enterprise data with diverse characteristics in different ways.⁷⁵ Core functions of enterprise applications, e. g. accessing relationally structured transactional data, can be executed through SQL. However, as more and more applications involve semi-structured, unstructured or text data, the SAP HANA database provides an appropriate engine (that is, a *text search engine*) in addition to its traditional relational query engine.⁷⁶ Also, a *graph engine* is provided, that is able to run graph algorithms on networks of data entities in order to support application functionalities such as production planning, supply chain optimization or social network analyzes.⁷⁷
- *Pushing application logic to the database layer:*
An application that executes the application logic on a certain data set needs to get the data from the database, process it, and possibly write the results back to the database.

⁷² The SAP HANA Studio is an Eclipse-based administration and data-modeling tool for SAP HANA.

⁷³ Cf. Vey, G./Krutov, I. (2012), p. 17

⁷⁴ Cf. Word, J. (2012), p. 30

⁷⁵ Cf. Faerber, F. et al. (2011), p. 46

⁷⁶ Cf. ibidem, p. 46

⁷⁷ Cf. ibidem, p. 46

This process involves a lot of communication overhead and latency and is also limited by the throughput of the network the communication is based on.⁷⁸ SAP HANA is able to push application semantics into the data management layer. Thus, the amount of data transfer can be reduced significantly and the overall processing becomes more efficient. SAP HANA also allows access to specific business logics implemented directly within the database engine.⁷⁹

- *Efficient communication with the application:*

In order to communicate efficiently with the application layer, SAP HANA plans to provide shared-memory communication with SAP proprietary application servers in order to more closely align the data types with each other.⁸⁰

⁷⁸ Cf. Vey, G./Krutov, I. (2012), p. 13

⁷⁹ Cf. Faerber, F. et al. (2011), p. 46

⁸⁰ Cf. ibidem, p. 46

5 Comparison of SAP HANA with RDBMS

In the following chapter SAP HANA will be compared to the relational database, which has been used for many years. The intention is to provide the reader with a brief, practical related comparison of the key differences between the two approaches. The differences will be explained in paragraphs and finally be illustrated in a table.

5.1 Hardware requirements

Hardware requirements of RDBMS

Concrete statements about the hardware requirements can only be done if enough customer information is available, e.g. amount of data, kind of data (backup, relevant, business related, etc.), data queries, data updates, security issues, etc..

Second, the requirements depend on the DBMS software chosen. Detailed information cannot be provided as these details are not available and usually be discussed by the customer and the vendor in several meetings.

To look at the operating system, as another requirement, as well, the “following table lists the supported server databases for each operating system.”⁸¹

Operating system	Supported database servers
AIX®	<ul style="list-style-type: none"> • DB2¹ 8.1, 8.2, 9 • Informix® 9.4 • Oracle³ 10g, 11g • Sybase 12.5.4
HP-UX	<ul style="list-style-type: none"> • DB2¹ 8.1, 8.2, 9 • Informix 9.4 • Oracle³ 10g, 11g • Sybase 12.5.4
Linux on Intel™	<ul style="list-style-type: none"> • DB2¹ 8.1, 8.2, 9 • Informix 9.4 • Oracle³ 10g, 11g • Sybase 12.5.4
Linux on zSeries	<ul style="list-style-type: none"> • DB2¹ 8.1
Solaris on Sparc	<ul style="list-style-type: none"> • DB2¹ 8.1, 8.2, 9 • Informix 9.4 • Oracle³ 10g, 11g • Sybase 12.5.4

⁸¹ w.a. (w. y. e)

Solaris on Intel	<ul style="list-style-type: none"> • Oracle³ 10g
Windows®	<ul style="list-style-type: none"> • DB2¹ 8.1, 8.2, 9 • Informix 9.4 • Microsoft® SQL Server 2000 SP4, 2005 SP1, 2008 • Oracle³ 10g, 11g • Sybase 12.5.4
z/OS	<ul style="list-style-type: none"> • DB2² 7
<p>1 Support for Enterprise Edition (EE), Enterprise-Extended Edition (EEE), and Workgroup Edition.</p> <p>2 Only the DB2® server is supported on z/OS® operating systems.</p> <p>3 Support for Enterprise Edition and Standard Edition.</p>	

Table 2 Supported server databases for different operating systems⁸²

Hardware requirements of SAP HANA

SAP partners with various hardware vendors that provide the hardware infrastructure required to run the SAP HANA software. The list of vendors includes but is not limited to: IBM, HP, Dell, Cisco, Fujitsu, NEC and Hitachi.⁸³ SAP HANA server offerings are named like the t-shirt sizes ranging from Extra-Small (XS) (128GB RAM) up to Extra Large (XL) (>2TB RAM).⁸⁴

All vendors offering hardware for HANA have to be certified for specific configurations and performance requirements.⁸⁵ In order to provide a common platform across all hardware vendors, the configurations have to fulfil the following requirements:⁸⁶

- The range of Intel Xeon processors that can be deployed is limited. From the currently available Intel Xeon processor E7 family, only the models E7-2870, E7-4870, and E7-8870 are possible processors. Moreover, the Intel Xeon processor X7560 can be deployed.
- SAP defines a certain main memory per core ratio to balance CPU processing power and the amount of data being processed. All configurations have to provide this ratio.
- All configurations have to meet certain performance requirements for diverse load profiles.

⁸² Cf. w. a. (w. y. c)

⁸³ Cf. Word, J. (2012), p. 33

⁸⁴ Cf. ibidem, p. 33

⁸⁵ Cf. Vey, G./Krutov, I. (2012), p. 18

⁸⁶ Cf. ibidem, p. 18

- The capacity of the storage devices that are used in the configurations have to meet certain sizing rules [see redbooks p. 26 ff.]
- The networking capabilities of the configurations have to include 10 gigabytes Ethernet.

Hardware that fulfils the requirements stated above will provide a well-performing infrastructure architecture for the deployment of SAP HANA.

The hardware vendor provides factory pre-installation for the hardware, the OS, and the SAP software. The installation is finalized with on-site setup and configuration of the SAP HANA components, which includes the deployment in the customer data center, setting up the connectivity to the network, the Solution Manager setup and other configurations. The customer can then build the connectivity to the source systems and clients, including the deployment of additional replication components on the source system.⁸⁷

5.2 Management of data objects

A relational database can manage data storage very flexibly as they are hold in simple structured tables. Complex data objects are separated into many simple data objects which are saved in tuples and columns. They are accessible through the relations and changes can be done in an easier way. SQL is the standard query language for relational databases.⁸⁸

Database index

In terms of using indices to process queries faster, it can be stated that there are no considerable differences between the ways SAP HANA and relational databases use indexing. While relational databases use indices in a row-based fashion, SAP HANA additionally uses compression techniques to accelerate data processing as described in section X. Dictionaries are used to match column attributes with integer numbers. These integer numbers replace the original text values in the according tables and serve as references to the actual text values. This can be considered as an additional way of indexing.⁸⁹

Row-based and column-based storage

See section 3.2 in sub-section *columnar and row-based storage*.

⁸⁷ Cf. Word, J. (2012), p. 33

⁸⁸ Cf. w. a. (w. y. c)

⁸⁹ Cf. Steilmann, R. (2012)

In terms of data organization forms, SAP HANA is considered as a “hybrid” database that uses both methods simultaneously to provide an optimal balance. The SAP HANA database allows the application developer to choose whether a table is to be stored row- or column-based. It also allows the developer to modify an existing table from columnar to row-based and vice versa. The decision to use columnar or row-based tables is typically determined by how the data is used and which method is the most efficient for that type of usage.⁹⁰

Most RDBMS store row-based whereas some RDBMS are also able to store column-based.

5.3 Programming interfaces

Programming interfaces in RDBMS

The query language used to access and modify data in RDBMS is SQL.

Programming interfaces in SAP HANA

SAP HANA uses SQL, SQLScript and Multidimensional Expressions (MDX) as programming interfaces.

- SQL

SQL is the main interface for client applications. The SQL implementation of the SAP HANA database is based on SQL 92 and SQL 99, but also contains various extensions on top.⁹¹ Some extensions regard the organizations of tables as both row-based and columnar tables and the conversion between the two formats.

- SQLScript

SQLScript is the own scripting language of the SAP HANA database. It mainly refers to functions that allow application-specific calculations to run inside the database.

- MDX Interface

MDX, which is considered the standard language for multidimensional queries, is used to connect clients (e. g. Microsoft Excel) and analytics applications.

⁹⁰ Cf. Word, J. (2012), p. 27

⁹¹ Cf. *ibidem*, p. 29

5.4 Data persistence

Data persistence in RDBMS

As data is stored and processed directly on the disk, data persistence is not a problematic issue in RDBMS and does not need to be further evaluated at this point.

Data persistence in SAP HANA

As main memory is only volatile storage, it is crucial to consider what will happen in case of a loss of power or any system failure.⁹²

The concept of a transaction should be evaluated at this point. A transaction is a fixed sequence of actions with a well-defined beginning and a well-defined ending. [61] This concept describes the characteristics of a transaction, also known under the term ACID – that is, Atomicity, Consistency, Isolation, and Durability.

Database managements systems supporting ACID transactions ensure that database transactions are processed efficiently and reliably.

The requirement “Durability” cannot simply be fulfilled by storing data in main memory as main memory is volatile storage, i. e. the data will be lost in case of a loss of electrical power. In order to ensure “Durability” – that is, make data persistent – it needs to be stored in a non-volatile way, e. g. on a hard disk or flash device.

The database storage is divided into pages. Whenever data is modified, changes to the corresponding pages will be marked and written to non-volatile storage. This is done in regular intervals. Meanwhile, each committed change made by a transaction is logged and written to persistent storage. Thus, durability can be guaranteed.

For example, SAP HANA stores modified pages in so-called save-points that are asynchronously written to non-volatile storage in defined intervals. Changes are logged synchronously. Thus, a log entry has to be written to non-volatile storage before the transaction can return and continue being processed.⁹³

The following figure shows the data save-points and logs in SAP HANA.

⁹² Cf. Vey, G./Krutov, I. (2012), p. 9

⁹³ Cf. ibidem, p. 9

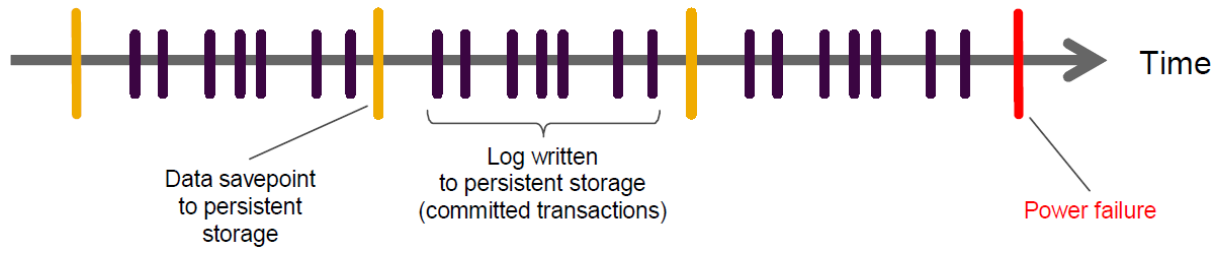


Figure 10 Save-points and logs in SAP HANA⁹⁴

⁹⁴ Contained in: *ibidem*, p. 9

5.5 Summary and additional aspects

The following table summarizes the comparison of SAP HANA and RDBMS with respect to several criteria. Furthermore, some additional criteria have been evaluated and results provided in the table, which have not been elaborated in this paper in detail.

<u>Aspect</u>	<u>RDBMS</u>	<u>SAP HANA</u>
Storage type	Hard Disc	Main memory
(Hardware) requirements	Depends on individual customer situation, standard components/common OS supported	Hardware is provided by external vendors who need to be certified according to specific minimum performance requirements.
Handling data objects	Traditionally row-oriented/column-oriented available, using related tables	Both row-based and columnar are supported
Query languages	SQL	SQL, SQLScript, MDX
Using indices	Yes	Yes; in addition: compression techniques leveraged
Locking	Optimistic/pessimistic, 2-Phase-Protocoll	Multiversion concurrency control (MVCC)
Data persistence	Hard disc	Writing log entries and page modifications to persistent disk volumes (hard disc/SSD); Loss of committed data possible as actual data processing occurs in main memory; However, easy recovery possible due to change log.
Data access times	Appropriate for querying transactional data	Fast. Appropriate for analytical processing
Parallel data processing	Parallelization across multiple cores, but not as efficient as in in-memory	Parallelization across multiple cores and machines. Data to be processed can be distributed across several servers/applications
Automatic Storage Management (ASM)	Volume Management integrated	Not available
Backup/Recovery	Various backup features, full data recovery from disc guaranteed	Recovery from disc according to save points and logging. Data loss in case of server crash or power loss
ACID compliance	Fully compliant	“Durability” can be ensured by writing changes to non-volatile storage in regular intervals
Experience/References	Established / existent since 1960s	Still in early stages / existent since 2010
Relative costs	Low	High

Table 3 Summary of the comparison of RDBMS and SAP HANA

6 Conclusion

In this paper, the controversially discussed technology of in-memory data management as well as SAP HANA as a proprietary prototype were to be compared with relational database management systems.

While doing research on in-memory computing and SAP HANA, the authors noticed that papers both from research and industry often adopt an either positive or negative attitude towards in-memory computing, respectively SAP HANA. This made the authors examine the literature with certain distance and posed the additional challenge to evaluate in-memory and relational databases in a neutral way.

Following key findings could be attained based on previous analyses:

In-memory data management and processing has many advantages compared to relational databases, such as:

- High data access speed through storage in main memory, columnar data storage and data compression techniques;
- Increased performance in data processing through pushing application logic to database level and through efficient communication between database and applications;
- Increased performance through parallelization of data processing across multiple cores and machines;
- Responding to changed business requirements like managing enterprise data with different characteristics through multi-engine query processing.

However, the most benefit from in-memory computing can be achieved when data need to be accessed very frequently and large amounts of data need to be analyzed. Thus, a decision maker from customer side should always consider the customer's individual situation and evaluate if in-memory technology is really required. Today, a main reason not to switch to in-memory technology is the cost aspect: In-memory computing, even though prices have decreased significantly in the last years, is still expensive compared to classical databases. Moreover, in-memory computing is still inexperienced and requires big efforts for implementation as well as the appropriate know-how.

SAP HANA as a proprietary system needs to be evaluated for every individual situation, especially with regards to its relatively high costs. On the other hand, there is a wide range of Open Source solutions and other alternatives that should be taken into consideration.

As it always depends on the individual customer situation whether or not in-memory computing has a high or low or even no value add to the business,

Finally, it can be concluded that even though in-memory computing delivers many benefits, an evaluation of how high the value add to the business is always needs to be considered under the cost aspect, i. e. a return on investment calculation for every individual customer situation.

Lists of references

List of literature

- Derwand, G. et al. (2013) High-end customer landscapes based on SAP HANA, Technical White Paper
- Elmasri, R./Navathe, S. (2007) Grundlagen von Datenbanksystemen, München: Pearson Education Deutschland GmbH
- Faerber, F. et al. (2011) SAP HANA Database, Data Management for Modern Business Applications, SIGMOD Record, Vol. 40, No. 4, p. 45-51
- Krueger, J. et al. (2010) Hauptspeicherdatenbanken für Unternehmensanwendungen, Datenmanagement für Unternehmensanwendungen im Kontext heutiger Anforderungen und Trends, in: Datenbank Spektrum, Vol. 10, p. 143-158
- Krueger, J. et. al. (2011) Main Memory Databases for Enterprise Applications
- Leve, J. (2007) Process Automation Handbook, London: Springer
- Nieendick, M./Jansen, J./Kalinowski, T. (2013) Big Data Management auf Basis von In-Memory-Technologien, in: Digitalisierung und Innovation, (Keuper, F. et al.), Wiesbaden: Springer Fachmedien
- Ott, J./Stirnemann, R. (2012) In-Memory-Datenbanken im Vergleich, in: Computerwoche, No. 3, p. 24-27
- Plattner, H./Zeier, A. (2012) In-Memory Data Management, Technology and Applications, 2nd Edn., Heidelberg et al.: Springer-Verlag
- Revesz, P. (2010) Introduction to Databases: From Biological to Spatio-Temporal, London: Springer
- Rohrmoser, K. (w. y.) Aktuelle Trends aus Business Intelligence & Datawarehouse, Königstein im Taunus: data2fact
- SAP (Publ.) (2012) SAP HANA Database for Next-Generation Business Applications and Real-Time Analytics, Explore and Analyze Vast Quantities of Data from Virtually Any Source at the Speed of Thought, White Paper
- Vey, G./Krutov, I. (2012) SAP In-Memory Computing on IBM eX5 Systems, w. l.: w. a.
- Word, J. (2012) SAP HANA Essentials, w. l.: Epistemy Press LLC

List of Internet and Intranet Resources

Petty, C. (2007)	Gartner Says Worldwide Relational Database Market Increased 14 Percent in 2006, http://www.gartner.com/newsroom/id/507466 , retrieval: 07/01/2013
Stelmann, R. (2012)	What are database Indexes?, http://www.linux.org/article/view/what-are-database-indexes- , retrieval: 06/21/2013
w. a. (2006)	Optimistic locking, http://pic.dhe.ibm.com/infocenter/wchelp/v6r0m0/index.jsp?topic=%2Fcom.ibm.commerce.admin.doc%2Fconcepts%2Fcpmoptlock.htm , retrieval: 06/29/2013
w. a. (2009)	Databases – Database Theory, http://www.montan.net/portal/Datenbanken_1_Datenbankarten.php , retrieval: 06/21/2013
w. a. (2010)	2-Phase-Locking, http://wikis.gm.fh-koeln.de/wiki_db/Datenbanken/2-Phasen-Sperrprotokoll , retrieval: 06/24/2013
w. a. (2012)	Software Stack Market Share, August 2012, http://www.tuicool.com/articles/qYR7N3 , retrieval: 06/29/2013
w. a. (2013a)	SAP HANA, http://db-memory.blogspot.de/2013/01/sap-hana.html , retrieval: 06/29/2013
w. a. (2013b)	Who Leads the RDBMS Pack?, http://databases.about.com/library/weekly/aa060401a.htm , retrieval: 07/01/2013
w. a. (w. y. a)	Databases, http://www.itwissen.info/definition/lexikon/Datenbank-DB-database.html , retrieval: 06/21/2013
w. a. (w. y. b)	Database – What is a database?, http://www.datenbanken-verstehen.de/datenbank-grundlagen/datenbank.php , retrieval: 06/22/2013
w. a. (w. y. c)	Optimistic and pessimistic locking, http://docs.jboss.org/jbossas/docs/Server_Configuration_Guide/4/html/TransactionJTA_Overview-Pessimistic_and_optimistic_locking.html , retrieval: 06/29/2013
w. a. (w. y. d)	RDBMS, http://www.itwissen.info/definition/lexikon/relational-database-management-system-RDBMS-Relationales-Datenbank-Managementsystem.html , retrieval: 06/22/2013
w. a. (w. y. e)	2-Phase-Locking, http://docs.oracle.com/cd/E17275_01/html/programmer_reference/lock_twopl.html , retrieval: 06/24/2013
w. a. (w. y. f)	2-Phase-Locking, http://research.microsoft.com/en-us/people/philbe/chapter3.pdf , retrieval: 06/24/2013
Webb, M. (w. y.)	Introduction to databases, http://falk-weihmann.de/Access/allg.htm , retrieval: 06/21/2013

INTRODUCTION TO MONGODB

Seminar Paper

provided on July 2, 2013

School of: Business

Program: International Business Information Management

Course: WWI2010I

by

Rocco Schulz and Robert Wawrzyniak

Baden-Wuerttemberg Cooperative State University Stuttgart

Contents

List of Abbreviations	III
List of Tables	III
List of Listings	IV
1 Introduction	1
1.1 Objectives	1
1.2 Methodology and Structure	1
2 Foundation	3
2.1 NoSQL Databases	3
2.2 MongoDB	4
3 Setting up MongoDB	6
3.1 Installation	6
3.2 Configuration	6
4 Using MongoDB	8
4.1 Test Environment Setup	8
4.2 Command Line Interface	8
4.2.1 Read	8
4.2.2 Create	12
4.2.3 Update	14
4.2.4 Delete	16
4.3 Programmatically	16
5 Maintaining MongoDB	19
5.1 Monitoring	19
5.2 Security and Authentication	19
6 Conclusion and Outlook	22
Appendices	23
Lists of References	27

List of Tables

Tab. 1:	SQL terms mapped to MongoDB	5
Tab. 2:	Query selectors for value comparison	11

List of Listings

1	Commands to set up the MongoDB environment on Windows	6
2	Shell commands to download and load the enron email corpus into MongoDB on Linux	8
3	Command to get a list of available collections in the current database	9
4	<code>findOne</code> query on the messages collection	9
5	Calculating the percentage of replies	10
6	Array queries on an inventory collection	10
7	Exemplary usage of query operators	10
8	Trying to sort the messages collection by a non-indexed field	11
9	Showing all indices of the messages collection	12
10	(Bulk) Insertion of Documents	13
11	Updating MongoDB Collections	14
12	Removing Documents from a MongoDB Collection	16
13	Installing the mongo gem	17
14	Installing the Mongoid gem	17
15	Exemplary Usage of Mongoid	18
16	Output of mongotop	19
17	Creating an administration user	20
18	Adding a user to the enron database	21
19	Sample MongoDB configuration file	25
20	Example mongoid.yml	26

1 Introduction

The recent growth of user-driven content as well as the availability of sensors are accounting for huge amounts of growing data which are becoming increasingly less structured. Handling this semi-structured data has introduced new challenges for classical processing tools such as relational database management systems because upfront schema definitions are no longer practical in this environment.¹ A number of data stores have evolved that address specific issues related to the processing of big data. These storage systems, commonly subsumed under the term NoSQL have become increasingly popular in web applications that need to operate reliably under extreme load in near real-time. These usage scenarios have influenced other areas so that new software patterns were developed resulting in new software products suited for these types of situations.² Many of these products are open source and freely available for anyone to use. MongoDB, which is a document-oriented database, is one of these open source applications that has become very popular over the past years. A reason for the high popularity is the availability of cheap hardware, which allows access to large scale data processing for a broad audience. High prices for professional data software are a strong motivator to turn to open source alternatives such as MongoDB.³

1.1 Objectives

This paper should familiarize the reader with the basic usage of the document-oriented database MongoDB. Create, read, update and delete (CRUD) operations as well as the initial setup of MongoDB should be covered.

Additionally deployment-related topics such as configuration, security concerns and monitoring should be covered to allow a first evaluation for production usage.

This paper does not claim to be a complete guide to MongoDB but rather tries to provide insights into MongoDB's capabilities.

1.2 Methodology and Structure

Section 2 first provides a brief introduction to the bigger context of NoSQL databases before explaining the concepts and exemplary use cases of MongoDB as one representative of NoSQL databases.

The following sections focus on the usage of MongoDB. Section 3 covers the installation and

¹ Cf. Tiwari, S. (2011), p. XVII

² Cf. Warden, P. (2011), p. VII

³ Cf. *ibid.*, p. VII

basic configuration of MongoDB and section 4 shows how to run basic create, read, update and delete operations (CRUD) on MongoDB in different environments such as the mongo shell, with graphical tools or programmatically via a driver. Most examples are based on a publicly available data set consisting of several hundred thousand email messages to allow reproducibility of exemplary results.

Section 5 completes the introduction to the topic by providing essential information on monitoring and other maintenance related concerns. Section 6 finally concludes and gives a thorough outlook on advanced topics and future developments of MongoDB.

2 Foundation

2.1 NoSQL Databases

NoSQL describes a range of data-store technologies that do not have an SQL interface. There are multiple types of NoSQL data stores, these are briefly introduced in the following sections.

Key-Value Stores As the name implies a key-value store maps keys to values in a similar way as a map or hash-table would in a high level programming language.⁴ Such data structures are very efficient with an average algorithm running time of big $O(1)$ for accessing data and are hence often used for caching of data.⁵ However they are not very useful when it is necessary to do queries that are more complex than simply looking up a value for a given key.

There are various open source projects available, whereas memcached, Voldemort, Redis and Riak belong to the more popular ones.⁶

Document Oriented Databases Document oriented databases store documents which can contain a variety of values which may be of any type, including sub-documents, so that nested structures can be created.⁷

These databases should not be confused with document management systems. Even though binary documents can be stored as well, the word document in this context refers to a structured set of keys and values that belong to a document.⁸

Multiple documents form collections which can be queried. Document oriented databases do not have a fixed schema and hence exhibit high flexibility.

Document oriented databases are not a new phenomenon as they have been around for a few decades already. However recently a number of new open source document oriented databases like MongoDB and CouchDB have quickly become popular.⁹

Graph Databases Graph databases are an excellent choice when dealing with highly inter-connected data. They consist of multiple nodes and relationships between these nodes. Nodes as well as their relationships can contain additional information in the form of key-value pairs.¹⁰

⁴ Cf. Redmond, E./Wilson, J. R. (2012), p. 4

⁵ Cf. Tiwari, S. (2011), p. 14 et sq.

⁶ Cf. Redmond, E./Wilson, J. R. (2012), p. 4

⁷ Cf. *ibid.*, p. 5 et sq.

⁸ Cf. Tiwari, S. (2011), p. 18 et sq.

⁹ Cf. *ibid.*, p. 18 et sq.

¹⁰ Cf. Redmond, E./Wilson, J. R. (2012), p. 6

Graph databases facilitate rapid traversal through graphs by following relationships. Well known representatives of this group of NoSQL databases are Neo4j¹¹ and FlockDB^{12, 13}

2.2 MongoDB

“MongoDB is designed to be huge (the name mongo is extracted from the word humongous). Mongo server configurations attempt to remain consistent if you write something, subsequent reads will receive the same value (until the next update). This feature makes it attractive to those coming from an RDBMS background. It also offers atomic read-write operations such as incrementing a value and deep querying of nested document structures. Using JavaScript for its query language, MongoDB supports both simple queries and complex map-reduce jobs.”¹⁴

MongoDB belongs to the group of document oriented NoSQL databases and is written in C++. It does not try to be a general purpose database but has been developed with the philosophy “One size does not fit all” in mind.¹⁵

MongoDB differs from other document-oriented databases in its ability to easily scale across several servers. This is possible with replication, where data is copied to other servers to provide redundancy or by sharding where large collections are split into pieces to be distributed among all nodes.¹⁶

Terminology The MongoDB terminology differs from the terms used in RDBMS due to the different concepts. MongoDB works with documents instead of tables as opposed to conventional RDBMS. A document is a single storage unit and multiple documents can be contained in a collection. Documents are stored in BSON format which is a JSON based specification.¹⁷

Most terms can roughly be mapped to their SQL counterparts. An overview of key terms is provided in Tab. 1.

Areas of application A design goal of MongoDB is high availability through replication.¹⁹ MongoDB aims for good horizontal scalability, speed and simplicity. Transaction support has been omitted as a trade-off in favor of these attributes.²⁰

MongoDB is hence a good fit for applications that require a fast and highly available data store,

¹¹ See <http://www.neo4j.org/>

¹² See <https://github.com/twitter/flockdb>

¹³ Cf. Tiwari, S. (2011), p. 19

¹⁴ Redmond, E./Wilson, J. R. (2012), p. 6

¹⁵ Cf. Plugge, E./Hawkins, T./Membrey, P. (2010), p. 3

¹⁶ Cf. Redmond, E./Wilson, J. R. (2012), p. 165

¹⁷ Cf. Pasette, D. et al. (2013)

¹⁹ Cf. Plugge, E./Hawkins, T./Membrey, P. (2010), p. 3

²⁰ Cf. *ibid.*, p. 5

SQL Terms/Concepts	MongoDB Terms/Concept
database	database
table	collection
row	document
column	field
index	index
table joins	embedded documents and linking
primary key	primary key
Specify any unique column or column combination as primary key.	In MongoDB, the primary key is automatically set to the <code>_id</code> field.

Tab. 1: SQL terms mapped to MongoDB¹⁸

which is one of the reasons why it is very popular for web applications. Due to its document-oriented nature MongoDB is a good fit for hierarchical data such as category hierarchies and comments.

In MongoDB comments can be contained in their parent document as a sub-collection. Support for threaded comments can be added by embedding comments in their parent comment. As a result the whole comment thread can be obtained by simply retrieving the embedded collection of comments from an item. In a relational database this could only be achieved by more sophisticated queries that require joins.

As opposed to RDBMS, MongoDB does not have fixed formats for the items that are to be stored. Differed types of objects with different sets of attributes can easily be put into the same collection.

E-Commerce product catalogs must have this capability in order to store information of various articles. In relational databases this capability could be provided via multiple table inheritance. While this allows flexible structures, it also requires expensive join operations to retrieve all relevant attributes of a product. "MongoDBs dynamic schema means that each document need not conform to the same schema. As a result, the document for each product only needs to contain attributes relevant to that product."²¹

²¹ 10gen, Inc. (2013h)

3 Setting up MongoDB

3.1 Installation

Windows The latest version of MongoDB can be downloaded from <http://www.mongodb.org/downloads>. After downloading, the archive needs to be extracted, e.g. to `C:\mongodb-win32-x86_64-[version]`. Afterwards the environment needs to be set up by running the following sequence of commands in the Windows Command Prompt.

```
cd \
:: rename folder
move C:\mongodb-win32-* C:\mongodb
:: create folder for data in default location
md data
md data\db
```

Lst. 1: Commands to set up the MongoDB environment on Windows

MongoDB can now be started by issuing the command `C:\mongodb\bin\mongod.exe`. Since version 2 it is also possible to run `mongod` as a service on Windows.²² Refer to the online documentation for instructions on how to do this.²³

Linux On Linux MongoDB can be installed via the integrated package manager. Most distributions do not ship the latest version of MongoDB. This paper describes the usage of MongoDB version 2.4 which can be obtained from 10gen's own repository if not available in the Linux distribution. See <http://docs.mongodb.org/manual/installation/>

3.2 Configuration

There are two configuration interfaces for MongoDB: The command line and the configuration file. When installed on Linux, MongoDB comes with a default configuration file under `/etc/mongodb.conf` which is shown in appendix 1 for convenience.²⁴ This file will be used automatically when `mongod` is run as a service, but needs to be specified explicitly when starting `mongod` via the `mongod` command (`mongod -f /etc/mongod.conf`).

In order to start MongoDB with a different data storage path on port 20000, the following command could be issued: `mongod -dbpath /media/mongodisk/data -port 20000`. Additional

²² Cf. 10gen, Inc. (2013f)

²³ See <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-windows/>

²⁴ Cf. 10gen, Inc. (2013b)

options are available as documented in the manual.²⁵ Section 5.2 also shows some security related configurations.

²⁵ See <http://docs.mongodb.org/manual/administration/configuration/>

4 Using MongoDB

4.1 Test Environment Setup

The following usage scenarios will be based on a publicly available data-set. This section describes how to obtain and load this data-set into the previously installed MongoDB.

The Enron Email Corpus has been ported to MongoDB and can be obtained from <http://mongodb-enron-email.s3-website-us-east-1.amazonaws.com/>. The uncompressed data-set contains approximately 500,000 emails. While this is a relatively small data-set it is sufficient for testing basic features of the database.

Once the archive has been downloaded and unpacked it can be loaded into MongoDB by using the command line tool `mongorestore`.²⁶ On Linux, run the commands as shown in Lst. 2.

```
wget https://s3.amazonaws.com/mongodb-enron-email/enron_mongo.tar.bz2
tar -xjvf enron_mongo.tar.bz2
mongorestore --db enron --noobjcheck dump/enron_mail/
```

Lst. 2: Shell commands to download and load the enron email corpus into MongoDB on Linux

Note that the loading process requires a running instance of the MongoDB server. Verify that the import was successful by opening the mongo shell and typing `show dbs`. The output should contain a row with `enron 3.9521484375GB`²⁷, which is the database that has been created during the loading process.

4.2 Command Line Interface

The command line interface for MongoDB is called the mongo shell and is the most powerful tool for operating on MongoDB databases. The shell language is JavaScript, so that own functions and variables can be declared within the shell. The following examples assume that the Enron email corpus has been loaded into the database `enron`. In order to start working on this database, type `use enron`.

4.2.1 Read

Before performing any read operations it is necessary to figure out what collections there are in the `enron` database, that can be read from. This information can be accessed via the command `show collections` as shown in Lst. 3

²⁶ Cf. 10gen, Inc. (2013g)

²⁷ The size of the database may differ due to different file systems

```
> show collections
messages
system.indexes
```

Lst. 3: Command to get a list of available collections in the current database

Making Queries In the mongo shell, the `find()` and `findOne()` functions can be used to query for data.²⁸ These queries are performed by calling them on any collection. Obtaining all emails from the messages collection is e.g. done by running `db.messages.find()`.

The syntax of the find command is as follows: `db.collection.find(<query>, <projection>)`

The `<query>` argument is used for filtering and the `<projection>` is used to specify or limit the fields that should be returned. The projection is hence comparable to the *SELECT* part of a SQL query and the query parameter is comparable to the *WHERE* clause of a SQL query.

The `findOne()` function is very similar, however it only returns the first element that matches the query. Lst. 4 shows the result of the `findOne` query on the messages collection without a query document provided. The second argument tells MongoDB to exclude the *body* field from the result. Setting the projection object to `{"subject":1}` would tell MongoDB to exclude all fields from the result except for *body*.

```
> db.messages.findOne({}, {"body": 0})
{
  "_id" : ObjectId("4f16fc97d1e2d32371003e27"),
  "subFolder" : "notes_inbox",
  "mailbox" : "bass-e",
  "filename" : "450.",
  "headers" : {
    "X-cc" : "",
    "From" : "michael.simmons@enron.com",
    "Subject" : "Re: Plays and other information",
    "X-Folder" : "\\Eric_Bass_Dec2000\\Notes Folders\\Notes inbox",
    "Content-Transfer-Encoding" : "7bit",
    "X-bcc" : "",
    "To" : "eric.bass@enron.com",
    "X-Origin" : "Bass-E",
    "X-FileName" : "ebass.nsf",
    "X-From" : "Michael Simmons",
    "Date" : "Tue, 14 Nov 2000 08:22:00 -0800 (PST)",
    "X-To" : "Eric Bass",
    "Message-ID" : "<6884142.1075854677416.JavaMail.evans@thyme>",
    "Content-Type" : "text/plain; charset=us-ascii",
    "Mime-Version" : "1.0"
  }
}
```

Lst. 4: `findOne` query on the messages collection

²⁸ Cf. w.a. (2013a), p. 7

As can be seen from the output in Lst. 4 there can also be nested documents within one document. In this case the headers field contains an object with all header fields. MongoDB allows querying for nested attributes as well. So in order to find out the percentage of replies among all emails we could run a series of commands as shown in Lst. 5.

```
> nAll = db.messages.count()
501513
> nReply = db.messages.find({"headers.Subject": /Re:.*\/}).count()
100195
> nReply / nAll * 100
19.978544923062813
```

Lst. 5: Calculating the percentage of replies in the messages collection within the mongo shell

Similarly it is possible to query for values which are stored in an array instead of a sub-document. If the array contains sub-documents it is also possible to query for specific fields within the sub-documents.²⁹ This is demonstrated in Lst. 6

```
//find documents with exactly this array in the tags attribute
db.inventory.find( { tags: ['fruit','food','citrus'] } )
//find documents that contain a 'fruit' element in their tags array
db.inventory.find( { tags: 'fruit' } )
//find documents that have 'fruit' as the first element in the tags array
db.inventory.find( {'tags.0': 'fruit' } )
//find documents where the memos field contains an array that contains at least one
//subdocument with the field 'by' with the value 'shipping'
db.inventory.find( {'memos.by': 'shipping' } )
```

Lst. 6: Array queries on an inventory collection

Besides using regular expressions as demonstrated in Lst. 5 MongoDB offers a number of other query selectors for comparisons (see Tab. 4).

An exemplary query using the greater than ($\$gt$), less than ($\lt) and or ($\$or$) operator is shown in Lst. 7. This query selects all documents in the inventory collection which have a type attribute with a value of 'food' and either have a quantity attribute with a value above 100 or a price attribute with a value below 9.95.

```
db.inventory.find( { type:'food', $or:[ { qty:{ $gt:100 } },{ price:{ $lt:9.95 } } ] } )
```

Lst. 7: Exemplary usage of query operators³⁰

²⁹ Cf. w.a. (2013a), p. 10

³¹ 10gen, Inc. (2013k)

Name	Description
\$all	Matches arrays that contain all elements specified in the query.
\$gt	Matches values that are greater than the value specified in the query.
\$gte	Matches values that are equal to or greater than the value specified in the query.
\$in	Matches any of the values that exist in an array specified in the query.
\$lt	Matches vales that are less than the value specified in the query.
\$lte	Matches values that are less than or equal to the value specified in the query.
\$ne	Matches all values that are not equal to the value specified in the query.
\$nin	Matches values that do not exist in an array specified to the query.

Tab. 2: Query selectors for value comparison³¹

Optimizing queries with indices When sorting a data-set MongoDB inspects all documents within the collection to put them in the desired order. In large collections this can be very time consuming and results in long response times. MongoDB actually refuses to run queries which would take too long to complete as shown in Lst. 8.

```
//find all emails and sort them ascending by the senders email address
> db.messages.find().sort({"headers.From":1})
error: {
  "$err" : "too much data for sort() with no index.  add an index or specify a smaller limit",
  "code" : 10128
}
```

Lst. 8: Trying to sort the messages collection by a non-indexed field

Just like in other databases indices can be created to improve the efficiency of read operations.³² Indices can be created using the `ensureIndex(<field1>: <order>, <field2>: <order>, ...)` function in the mongo shell.³³ The internal mechanisms behind indexing of MongoDB are beyond the scope of this paper and can be read about in the official documentation.³⁴ An index for the *From* field in the embedded *headers* document can be created with `db.messages.ensureIndex("headers.From":1)`, so that the query shown in Lst. 8 can be ex-

³² Cf. Redmond, E./Wilson, J. R. (2012), 152 et sq.

³³ Cf. w.a. (2013a), p. 12

³⁴ See <http://docs.mongodb.org/manual/core/indexes/>

ecuted. It is possible to display all current indices by executing `getIndices()` as shown in Lst. 9.

9.

```
> db.messages.getIndices()
[ {
  "v" : 1,
  "key" : { "_id" : 1 },
  "ns" : "enron.messages",
  "name" : "_id_"
},
{
  "v" : 1,
  "key" : { "headers.From" : 1 },
  "ns" : "enron.messages",
  "name" : "headers.From_1"
} ]
```

Lst. 9: Showing all indices of the messages collection

Further optimization strategies include query tuning, storing pre-calculated data and restructuring document formats and are described in detail in *50 Tips and Tricks for MongoDB Developers* by Chodorow, K.³⁵

4.2.2 Create

Documents can be created with the `insert()` command in the mongo shell.

Bulk inserts are also possible (see listing 10).³⁶

```
1 // switch to the store db
2 use store
3
4 // an object to save to the 'books' collection
5 > var book = {
6   _id : 1,
7   title : "MongoDB: The Definitive Guide",
8   author : "Kristina Chodorow",
9   publisher : "O'Reilly Media"
10 };
11
12 // an array of books to add to the collection
13 > var bookArr = new Array();
14
15 > bookArr.push({
16   _id : 2,
17   title : "MongoDB in Action",
18   author : "Kyle Banker",
19   publisher : "Manning Publications"
20 });
21
22 > bookArr.push({
23   _id : 3,
24   title : "MongoDB with Python and Ming",
```

³⁵ Cf. Chodorow, K. (2011)

³⁶ Cf. w.a. (2013a), p. 53 et sqq.

```
25   author : "Rick Copeland",
26   publisher : "Amazon Digital Services"
27 });
28
29 // continue adding objects to the array
30
31 // insert one object:
32 // -> SQL:      INSERT INTO TABLE books
33 //              VALUES _id=1, title="", author="", publisher=""
34 > db.books.insert(book);
35
36 // bulk insert:
37 > db.books.insert(bookArr);
```

Lst. 10: (Bulk) Insertion of Documents

4.2.3 Update

Updates can be done with the `update()` command and the `save()` command (see listing 11).³⁷

```

1 // syntax for updating:
2 > db.collection.update( <query>, <update>, <options> )
3
4 // update a field
5 // -> SQL:      UPDATE books
6 //              SET title="MongoDB: The Definitive Guide, 2nd Edition"
7 //              WHERE _id=1
8 > db.books.update(
9   { _id: 1 },
10  { $set: {
11    title : "MongoDB: The Definitive Guide, 2nd Edition"
12  } },
13  )
14
15 // add a new field to all documents
16 // -> SQL:      ALTER TABLE books
17 //              ADD price FLOAT
18 //              ADD currency VARCHAR(3)
19 > db.books.update(
20   {},
21   { $set: {
22     price: 9.99,
23     currency: 'USD'
24   }
25 }
26 )

```

Lst. 11: Updating MongoDB Collections

Name	Description
\$inc	Increments the value of the field by the specified amount.
\$rename	Renames a field.
\$setOnInsert	Sets the value of a field upon documentation creation during an upsert. Has no effect on update operations that modify existing documents.
\$set	Sets the value of a field in an existing document.
\$unset	Removes the specified field from an existing document.

Tab. 3: Field Update Operators³⁸

³⁷ Cf. w.a. (2013a), p. 75 et. sqq.

³⁸ 10gen, Inc. (2013m)

³⁹ 10gen, Inc. (2013m)

Name	Description
Operators	
\$	Acts as a placeholder to update the first element that matches the query condition in an update.
\$addToSet	Adds elements to an existing array only if they do not already exist in the set.
\$pop	Removes the first or last item of an array.
\$pullAll	Removes multiple values from an array.
\$pull	Removes items from an array that match a query statement.
\$pushAll	Deprecated. Adds several items to an array.
\$push	Adds an item to an array.
Modifiers	
\$each	Modifies the \$push and \$addToSet operators to append multiple items for array updates.
\$slice	Modifies the \$push operator to limit the size of updated arrays.
\$sort	Modifies the \$push operator to reorder documents stored in an array.

Tab. 4: Array Update Operators³⁹

4.2.4 Delete

Delete operations for one or multiple documents can be performed with `db.collection.remove()` (see listing 12).⁴⁰

```

1 // remove all documents where the title contains 'MongoDB'
2 // -> SQL:      DELETE FROM books
3 //             WHERE title LIKE '%MongoDB%'
4 //             (LIMIT 1)
5 > db.books.remove({
6   title : /MongoDB/i
7 })
8
9 // remove only one documents
10 > db.books.remove({
11   title : /MongoDB/i,
12   1 /* <justOne> = true*/
13 })
14
15 // clear a collection
16 // -> SQL:      DELETE * FROM books
17 > db.books.remove()
18
19 // drop the collection
20 // -> SQL:      DROP TABLE books
21 > db.books.drop()

```

Lst. 12: Removing Documents from a MongoDB Collection

4.3 Programmatically

In order to access MongoDB programmatically it is necessary to use a *driver* to connect to the database and to translate MongoDB constructs into appropriate ones for your target language. 10gen provides drivers for most languages (with additional ones developed by the community) including but not limited to:⁴¹

1. C/++/#
2. Java
3. JavaScript/Node.js
4. PHP
5. Python
6. Ruby

This section will show how to use MongoDB programmatically using the Ruby language.

⁴⁰ Cf. w.a. (2013a), p. 83 et. sqq.

⁴¹ Cf. 10gen, Inc. (2013e)

Installing the driver

```

1  #!/bin/bash
2
3  # update the rubygems package manager
4  gem update --system
5
6  # install the mongodb driver
7  gem install mongo
8
9  # install the C version of BSON to improve performance
10 gem install bson_ext

```

Lst. 13: Installing the mongo gem

The driver allows interacting with MongoDB through Ruby by employing similar constructs to the examples above using the Mongo shell, however most developers use an Object Document Mapper (ODM) to gain higher-level abstractions and modeling functionality.⁴² This paper will focus on Mongoid, one of the most popular ODMs.

Installing Mongoid

```

1  #!/bin/bash
2
3  gem install mongoid

```

Lst. 14: Installing the Mongoid gem

After installing the gem, it is necessary to configure Mongoid so that it connects to the correct MongoDB instance. Mongoid is configured by creating and modifying the file `mongoid.yml`, of which an example is located in appendix 2.

The exact mechanism of loading the configuration is dependent upon which framework (if any) is used by the developer. Proper guidance as to how to proceed can be found in the Mongoid documentation, this includes advanced features such as logging, sharding and replica sets.⁴³

Using Mongoid Listing 15 shows essential usage of the Mongoid ODM. In addition to the highlighted features, Mongoid allows idiomatic usage of every MongoDB feature. Thus the resulting feature set and documentation is quite extensive and is recommended to gain a deeper understanding, see <http://mongoid.org/en/mongoid/> for more details.

⁴² Cf. 10gen, Inc. (2013l)

⁴³ Cf. w.a. (2013b)

```
1  /* require the mongoid gem so that it can be used */
2  require mongoid
3
4  /* define a model class */
5  class Person
6    include Mongoid::Document
7    field :first_name, type: String
8    field :last_name, type: String
9    field :dateOfBirth, type: Date
10   field :job, type: String
11   field :company, type: String
12   field :salary, type: Integer
13 end
14
15 /* instantiate a Person */
16 john = Person.new(
17   first_name: John,
18   last_name: Doe,
19   date_of_birth: Date.new(1971, 04, 20),
20   job: "Assistant Manger",
21   company: "FooCorp",
22   salary: 5000)
23
24 /* persist the object to the db, raise errors if validation fails */
25 john.save!
26
27 /* retrieve all employees of FooCorp */
28 foos = Person.where(company: "FooCorp")
29
30 /* print each employees name */
31 foos.each |employee| do
32   print employee.first_name
33 end
```

Lst. 15: Exemplary Usage of Mongoid

5 Maintaining MongoDB

5.1 Monitoring

Monitoring is an elementary part of database administration in order to avoid incidents before they happen. MongoDB ships with separate tools for monitoring. One of these is `mongotop` which can be used to get performance information on collections level.⁴⁴

	ns	total	read	write
2013-06-25T19:28:48				
	enron.messages	1158ms	1158ms	0ms
	enron.system.indexes	0ms	0ms	0ms
	enron.system.namespaces	0ms	0ms	0ms
	enron.system.users	0ms	0ms	0ms
	local.startup_log	0ms	0ms	0ms
	local.system.indexes	0ms	0ms	0ms

Lst. 16: Output of `mongotop`

Another command line tool for monitoring is `mongostat`, which provides information on a higher level such as number of open connections, memory consumption, network traffic and query throughput.⁴⁵

Additionally it is possible to get these information via a web interface. “In default configurations the REST interface is accessible on 28017. For example, to access the REST interface on a locally running `mongod` instance: `http://localhost:28017`”⁴⁶. The REST interface is disabled by default and needs to be enabled by setting the `rest` field to `true` in the configuration file.⁴⁷

It is also possible to get these information from within the mongo shell via commands such as `db.serverStatus()`. Additionally profiling information can be obtained when the profiling is enabled. This can be done by setting the profiling level to 1 or 2 via `db.setProfilingLevel(<level>)`. When set to 1, all operations are profiled, when set to 2 only slow operations will be profiled. A full list of available commands is available in the online documentation.⁴⁸

5.2 Security and Authentication

It is recommended to run MongoDB in a trusted environment, to make sure that only trusted machines may access the MongoDB server.⁴⁹ If MongoDB is run on a server which is publicly

⁴⁴ Cf. 10gen, Inc. (2013i)

⁴⁵ Cf. *ibid.*

⁴⁶ *Ibid.*

⁴⁷ Cf. 10gen, Inc. (2013c)

⁴⁸ See <http://docs.mongodb.org/manual/reference/command/#diagnostic-commands>

⁴⁹ Cf. Chodorow, K./Dirolf, M. (2010), p. 118

accessible, it should be run with the `-bindip` option to specify a local IP address that MongoDB will be bound to (e.g. `-bindip localhost` to only allow connections from an application on the same machine).⁵⁰

Access restrictions can be put in place by using MongoDB's authentication system. Before using the authentication framework one should create an administration user with the commands shown in Lst. 17.⁵¹ Note that the creation of an administration user is only possible when working on localhost with MongoDB's localhost authentication bypass enabled, which allows full administrative access to local users. This setting should be disabled after the creation of an administration user via `mongod -setParameter enableLocalhostAuthBypass=0` during startup or by setting this option in the configuration file.

```
//switch to the admin DB
> db = db.getSiblingDB('admin')
admin
// create a superuser with the role "userAdminAnyDatabase"
> db.addUser({user:"root",pwd:"toor",roles: ["userAdminAnyDatabase"]})
{
  ^^I"user" : "root",
  ^^I"pwd" : "6a921fa21bbcd22989efecbcb2340d17",
  ^^I"roles" : ["userAdminAnyDatabase"],
  ^^I"_id" : ObjectId("51d16db53ac63115056d6341")
}
```

Lst. 17: Creating an administration user

Users of a database are stored as documents in the `system.users` collection within this database. User documents consist of a `user` field which contains the username, a `readOnly` flag and a `pwd` field which stores a hash of the username and password.⁵²

Authentication in MongoDB can be enabled by either starting `mongod` with the `-auth` switch or by configuring it in the settings file with the line `auth = true`.

Lst. 18 shows how a user can be added to the `enron` database and how this user can be used to authenticate in the shell.

⁵⁰ Cf. Chodorow, K./Dirolf, M. (2010), p. 118

⁵¹ Cf. 10gen, Inc. (2013d)

⁵² Cf. 10gen, Inc. (2013a)

```
> use enron
switched to db enron
> show collections
Mon Jul 1 14:11:13.055 JavaScript execution failed: error: {
  "$err" : "not authorized for query on enron.system.namespaces",
  "code" : 16550
} at src/mongo/shell/query.js:L128
//add a user with read and write access
> db.addUser({user:"mike", pwd:"abc123", roles: ["readWrite"]})
{
  "user" : "mike",
  "pwd" : "fbfddbdf3a7a9955a764d02b4c10fd84",
  "roles" : ["readWrite"],
  "_id" : ObjectId("51d169773ac63115056d6340")
}
//try to authenticate with wrong pw
> db.auth({user:"mike",pwd:"test"})
Error: 18 { code: 18, ok: 0.0, errmsg: "auth fails" }
0
> db.auth({user:"mike",pwd:"abc123"})
1
> show collections
messages
system.indexes
system.users
```

Lst. 18: Adding a user to the enron database with the auth option set to true

6 Conclusion and Outlook

MongoDB's strengths are its ability to handle large amounts of data as well as large amounts of requests by replication and sharding. Additionally it allows very flexible data models due to its schema-less storage model.

In section 4 it has been shown that MongoDB is easy to use as its query language shares many similarities with traditional SQL concepts.

MongoDB should not be seen as an alternative for conventional RDBMS but as a complementary module in application development. MongoDB encourages denormalization of schemas by not having any. This goes along with less constraints so that more caution is needed on the application development side.

There are various other features available in MongoDB which haven't been covered in this paper. These include auto-sharding, geospatial lookups, advanced optimization and clustering. Designing collections or document structures for MongoDB is also different to the design of relational database models and should be further looked into before considering the usage of MongoDB in production.

Appendices

List of Appendices

Appendix 1: MongoDB Configuration File	24
Appendix 2: Overview of the Mongoid Configuration File 'mongoid.yml'	26

Appendix 1: MongoDB Configuration File

```
1 # mongo.conf
2
3 #where to log
4 logpath=/var/log/mongod.log
5
6 logappend=true
7
8 # fork and run in background
9 fork = true
10
11 #port = 27017
12
13 dbpath=/var/lib/mongo
14
15 # location of pidfile
16 pidfilepath = /var/run/mongodb/mongod.pid
17
18 # Disables write-ahead journaling
19 # nojournal = true
20
21 # Enables periodic logging of CPU utilization and I/O wait
22 #cpu = true
23
24 # Turn on/off security. Off is currently the default
25 #noauth = true
26 auth = true
27
28 # Verbose logging output.
29 #verbose = true
30
31 # Inspect all client data for validity on receipt (useful for
32 # developing drivers)
33 #objcheck = true
34
35 # Enable db quota management
36 #quota = true
37
38 # Set oplogging level where n is
39 # 0=off (default)
40 # 1=W
41 # 2=R
42 # 3=both
43 # 7=W+some reads
44 #diaglog = 0
45
46 # Ignore query hints
47 #nohints = true
48
49 # Disable the HTTP interface (Defaults to localhost:27018).
50 #nohttpinterface = true
51
52 # enable rest access
53 rest = true
54
55 # Turns off server-side scripting. This will result in greatly limited
56 # functionality
57 #noscripting = true
58
59 # Turns off table scans. Any query that would do a table scan fails.
60 #notablesan = true
61
62 # Disable data file preallocation.
```



```
63 #noprealloc = true
64
65 # Specify .ns file size for new databases.
66 # nssize = <size>
67
68 # Account token for Mongo monitoring server.
69 #mms-token = <token>
70
71 # Server name for Mongo monitoring server.
72 #mms-name = <server-name>
73
74 # Ping interval for Mongo monitoring server.
75 #mms-interval = <seconds>
76
77 # Replication Options
78
79 # in replicated mongo databases, specify here whether this is a slave or master
80 #slave = true
81 #source = master.example.com
82 # Slave only: specify a single database to replicate
83 #only = master.example.com
84 # or
85 #master = true
86 #source = slave.example.com
```

Lst. 19: Sample MongoDB configuration file

Appendix 2: Overview of the Mongoid Configuration File 'mongoid.yml'

```

1 # Tell Mongoid which environment this configuration is for.
2 production:
3   # This starts the session configuration settings. You may have as
4   # many sessions as you like, but you must have at least 1 named
5   # 'default'.
6   sessions:
7     # Define the default session.
8     default:
9       # A session can have any number of hosts. Usually 1 for a single
10      # server setup, and at least 3 for a replica set. Hosts must be
11      # an array of host:port pairs. This session is single server.
12      hosts:
13        - flame.mongohq.com:27017
14      # Define the default database name.
15      database: mongoid
16      # Since this database points at a session connected to MongoHQ, we must
17      # provide the authentication details.
18      username: user
19      password: password
20      # This defines a secondary session at a replica set.
21      replica_set:
22        # This configuration is a 3 node replica set.
23        hosts:
24          - dedicated1.myapp.com:27017
25          - dedicated2.myapp.com:27017
26          - dedicated3.myapp.com:27017
27        database: mongoid
28        # We can set session specific options, like reads executing
29        # on secondary nodes, and defaulting the session to safe mode.
30        options:
31          consistency: :eventual
32          safe: true
33        # This defines a tertiary session at a Mongos fronted shard.
34        shard:
35          # This configuration is a Mongos shard server.
36          hosts:
37            - mongos.myapp.com:27017
38          database: mongoid
39          # This configuration shows an authenticated replica set via a uri.
40          another:
41            uri: mongodb://user:pass@59.1.22.1:27017,59.1.22.2:27017/mongoid
42      # Here we put the Mongoid specific configuration options. These are explained
43      # in more detail next.
44      options:
45        allow_dynamic_fields: false
46        identity_map_enabled: true
47        include_root_in_json: true
48        include_type_for_serialization: true
49        # Note this can also be true if you want to preload everything, but this is
50        # almost never necessary. Most of the time set this to false.
51        preload_models:
52          - Canvas
53          - Browser
54          - Firefox
55        scope_overwrite_exception: true
56        raise_not_found_error: false
57        skip_version_check: false
58        use_activesupport_time_zone: false
59        use_utc: true

```

Lst. 20: Example mongoid.yml

Lists of References

List of Literature

- Chodorow, K. (2011) 50 Tips and Tricks for MongoDB Developers, Sebastopol: O'Reilly Media
- Chodorow, K./Dirolf, M. (2010) MongoDB: The Definitive Guide, Sebastopol: O'Reilly Media
- Plugge, E./Hawkins, T./Membrey, P. (2010) The Definitive Guide to MongoDB: The NoSQL Database for Cloud and Desktop Computing, New York: Apress/Springer-Verlag
- Redmond, E./Wilson, J. R. (2012) Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement, Dallas: The Pragmatic Bookshelf
- Tiwari, S. (2011) Professional NoSQL, Indianapolis: John Wiley/Sons, Inc.
- Warden, P. (2011) Big Data Glossary, Sebastopol: O'Reilly Media

List of Internet and Intranet Resources

- 10gen, Inc. (2013a) MongoDB - Adding users, <http://docs.mongodb.org/manual/tutorial/add-user-to-database/>, retrieval: 07/01/2013
- 10gen, Inc. (2013b) MongoDB - Configuration, <http://docs.mongodb.org/manual/administration/configuration/>, retrieval: 07/01/2013
- 10gen, Inc. (2013c) MongoDB - Configuration File Options, <http://docs.mongodb.org/manual/reference/configuration-options/>, retrieval: 06/25/2013
- 10gen, Inc. (2013d) MongoDB - Creating a user administrator, <http://docs.mongodb.org/manual/tutorial/add-user-administrator/#localhost-exception>, retrieval: 07/01/2013

- 10gen, Inc. (2013e) MongoDB Drivers and Client Libraries, <http://docs.mongodb.org/ecosystem/drivers/>,
- 10gen, Inc. (2013f) MongoDB - Installation on Windows, <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-windows/>, retrieval: 06/26/2013
- 10gen, Inc. (2013g) MongoDB Manual - mongorestore, <http://docs.mongodb.org/manual/reference/program/mongorestore/>, retrieval: 04/22/2013
- 10gen, Inc. (2013h) MongoDB Manual - Use Cases - Product Catalog, <http://docs.mongodb.org/manual/use-cases/product-catalog/>, retrieval: 06/24/2013
- 10gen, Inc. (2013i) MongoDB - Monitoring, <http://docs.mongodb.org/manual/administration/monitoring/>, retrieval: 06/25/2013
- 10gen, Inc. (2013j) MongoDB - SQL comparison, <http://docs.mongodb.org/manual/reference/sql-comparison/>, retrieval: 04/20/2013
- 10gen, Inc. (2013k) Query Operators, <http://docs.mongodb.org/manual/reference/operator/#query-selectors>, retrieval: 05/24/2013
- 10gen, Inc. (2013l) Ruby Language Center, <http://docs.mongodb.org/ecosystem/drivers/ruby/>,
- 10gen, Inc. (2013m) Update – MongoDB Manual 2.4, <http://docs.mongodb.org/manual/core/update/>,
- Pasette, D. et al. (2013) BSON - Binary JSON, <http://bsonspec.org/>, retrieval: 05/29/2013
- w.a. (2013a) MongoDB CRUD Guide, <http://docs.mongodb.org/manual/MongoDB-crud-guide.pdf>, retrieval: 06/05/2013
- w.a. (2013b) Mongoid: Instalation, <http://mongoid.org/en/mongoid/docs/installation.html>,