



Jade Hochschule
Management, Information & Technologie
Wirtschaftsinformatik

Bachelorarbeit

über das Thema

**Prototypische Implementierung einer SAP UI5 Applikation
im SAP Umfeld und Analyse eines effizienten Einsatz von
UI-Objekten**

eingereicht von: Nils Lutz

bei: Prof. Dr. Hergen Pargmann
Prof. Dr. Harald Schallner

I Kurzfassung

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Abstract

Das ganze auf Englisch.

II Inhaltsverzeichnis

| | | |
|------------|--|-----------|
| I | Kurzfassung | I |
| II | Inhaltsverzeichnis | II |
| III | Abbildungsverzeichnis | IV |
| IV | Tabellenverzeichnis | V |
| V | Listing-Verzeichnis | V |
| VI | Abkürzungsverzeichnis | VI |
| 1 | Einleitung | 1 |
| 2 | Technologien | 2 |
| 2.1 | HTML5 | 2 |
| 2.2 | CSS3 | 8 |
| 2.3 | JavaScript | 10 |
| 2.4 | ABAP | 13 |
| 2.5 | SAP UI5 Framework | 13 |
| 2.5.1 | Definition | 13 |
| 2.5.2 | Architektur | 14 |
| 2.5.3 | OData Protokoll | 14 |
| 3 | Software Ergonomie | 15 |
| 3.1 | Definition | 15 |
| 3.2 | DIN EN ISO 9241 | 15 |
| 3.3 | Analyse Methoden | 16 |
| 3.4 | SAP Technologien in Bezug auf Software Ergonomie | 16 |
| 3.4.1 | Business Server Pages | 16 |
| 3.4.2 | Web Dynpro for ABAP | 16 |
| 3.4.3 | SAP Fiori / SAP UI5 / SAP Screen Personas | 16 |
| 4 | Fallbeispiel SAP UI5 | 17 |
| 4.1 | Beschreibung | 17 |
| 4.2 | Hilfsmittel | 17 |
| 4.2.1 | Entwicklungsumgebung | 17 |
| 4.2.2 | UI Design und Prototyping | 17 |
| 4.3 | Implementierung | 17 |
| 4.3.1 | View | 17 |
| 4.3.2 | Model und Controller | 19 |
| 4.3.3 | Backend | 20 |
| 5 | Analyse | 21 |
| 5.1 | Heatmap | 21 |
| 5.2 | UI-Objekte | 21 |

| | |
|-----------------------------|-----------|
| 5.3 PLATZHALTER | 21 |
| 6 Schluss | 22 |
| 7 Quellenverzeichnis | 23 |
| Anhang | I |
| A GUI | I |

III Abbildungsverzeichnis

| | | |
|--------|---|----|
| Abb. 1 | HTML5 Spezifikationen Übersicht | 3 |
| Abb. 2 | CSS-Boxmodell | 9 |
| Abb. 3 | DOM Beispielbaum | 12 |
| Abb. 4 | Model-View-Controller-Architekturmuster | 14 |

IV Tabellenverzeichnis

| | | |
|--------|---------------------------------------|---|
| Tab. 1 | HTML5 Browserkompatibilität | 4 |
|--------|---------------------------------------|---|

V Listing-Verzeichnis

| | | |
|---------|--|----|
| Lst. 1 | (X)HTML4.01 Doctype | 5 |
| Lst. 2 | HTML5 Doctype | 5 |
| Lst. 3 | HTML5 Basis Dokument | 5 |
| Lst. 4 | HTML5 Microdata | 6 |
| Lst. 5 | HTML5 Header und Footer | 6 |
| Lst. 6 | HTML5 Struktur Elemente | 6 |
| Lst. 7 | HTML5 Internet Explorer Fallback | 7 |
| Lst. 8 | HTML5 Formular Eingabe | 8 |
| Lst. 9 | CSS3 Syntax Beispiel | 8 |
| Lst. 10 | CSS3 medienspezifisches Stylesheet | 9 |
| Lst. 11 | CSS3 eigenschaftsspezifisches Stylesheet | 9 |
| Lst. 12 | Stylesheet Einbindung über Link Tag | 10 |
| Lst. 13 | Stylesheet Einbindung über Style Tag | 10 |
| Lst. 14 | Stylesheet Einbindung in HTML Tag | 10 |
| Lst. 15 | JavaScript Einbindung als separate Datei im Head | 10 |
| Lst. 16 | JavaScript Einbindung in Skript Tag im Head und Body | 11 |
| Lst. 17 | HTML5 Beispiel Definition | 11 |
| Lst. 18 | Root View der Applikation | 17 |
| Lst. 19 | Component.js - Datenmodell an die Root View binden | 19 |

VI Abkürzungsverzeichnis

| | |
|---------------|--|
| JSP | Java Server Pages |
| BSP | Business Server Pages |
| HTML | Hypertext Markup Language |
| CSS | Cascading Style Sheets |
| JS | JavaScript |
| WWW | World Wide Web |
| W3C | World Wide Web Consortium |
| XHTML | Extensible Hypertext Markup Language |
| XML | Extensible Markup Language |
| SGML | Standard Generalized Markup Language |
| DOM | Dokument-Objekt-Modell |
| WHATWG | Web Hypertext Application Technology Working Group |
| API | Application-Programming-Interface |

1 Einleitung

Motivation // wieso weshalb warum wo

// Beschreibung abatAG

// Entstehung des Projekts

Problemstellung // aktuelle situationsbeschreibung

// was soll besser laufen

Zielsetzung // Das Produkt - Template Programmierung für SAP Frontends mit SAP UI5

Struktur // der weg über die software ergonomie und ihre wichtigkeit, gezeigt über die Marktanalyse, hin zur praktischen Umsetzung durch Grundlagen und Beschreibung des Lösungsweges

2 Technologien

Zum besseren Verständnis der gesamten Thematik werden in den folgenden Kapiteln Technologien erläutert. Die Grundlagen und besonderen Merkmale der einzelnen Technologien helfen dabei die spätere Analyse nachvollziehen zu können. Zu den Kernsprachen, mit denen im Browser visuelle Informationen angezeigt und verändert werden können, zählen unter anderem die Auszeichnungssprache Hypertext Markup Language (HTML), die Gestaltungssprache Cascading Style Sheets (CSS) und die Skriptsprache JavaScript (JS). Aufbauend auf den drei genannten Sprachen setzen sich in der Regel Frameworks. Frameworks sind in sich konsistente Bibliotheken die gewisse Sprachkonstrukte, welche häufig benötigt werden in der Entwicklung, zur Verfügung stellen. Mit dem Einsatz eines Frameworks verfolgt man das Ziel oft geschriebenen Programmcode in eine Art „Bausatz-Konstruktion-Set“ auszulagern. So lässt sich ein einmal durchgeführter Entwicklungsprozess beliebig oft und mit weit weniger Aufwand bewerkstelligen, als wenn man jedes Mal den Programmcode von neuem entwickeln müsste.

2.1 HTML5

Historie HTML5 ist die aktuell empfohlene Spezifikation des World Wide Web Consortium (W3C) und sie stellt eine der Kernsprachen des World Wide Web dar. Angefangen hat es am 13. März 1989, als Tim Berners-Lee am CERN in Genf das World Wide Web (WWW) ins Leben gerufen und damit zusammen HTML festgelegt hat. So entstand ab 1990 eine Spezifikation seitens des W3C zur Festlegung und Vereinheitlichung der Kommunikation über das Internet. Im November 1995 erklärte das W3C HTML 2.0 zum offiziellen Sprachstandard. Grundlegende Unterschiede zwischen Version 1.0 und 2.0 existieren nicht. Version 3.0 der HTML Spezifikation ist gänzlich am Browser Markt vorbei definiert worden. Aus diesem Grund wurde HTML 3.2 ab Januar 1997 zum Nachfolger von Version 2.0 gemacht. Die folgende Entwicklung der Spezifikation brachte 1999 die überarbeitete Version 4.01 hervor. Im selben Zug wurde CSS, als Gestaltungssprache für HTML, immer mehr fokussiert. So begann die Fragmentierung der HTML Spezifikation und es existierten drei Versionen zur selben Zeit. Nämlich HTML 4.01 *strict*, die dem eigentlich definierten HTML am nächsten kam. HTML 4.01 *transitional*, nach welcher auch einige übliche physische Textauszeichnungen vorgesehen waren. „Physische Textauszeichnungen haben Bedeutungen wie „fett“ oder „kursiv“, stellen also direkte Angaben zur gewünschten Schriftformatierung dar. Bei physischen Elementen sollte der Web-Browser eine Möglichkeit finden, den so ausgezeichneten Text entsprechend darzustellen.“[Sela]. Sie wurde als Übergangslösung entwickelt. Die dritte Variante ist HTML 4.01 *frameset*. Der einzige Unterschied zur *transi-*

tional Variante ist, dass sich im Rumpf eines HTML Dokument ein Element verändert. Neben HTML wurde ab Januar 2000 auch eine Extensible Hypertext Markup Language (XHTML) genannte Spezifikation entwickelt, die HTML mit dem Extensible Markup Language (XML) Standard vereinen sollte. XHTML ist allerdings nicht als eigenständige Sprache zu verstehen, sondern als eine Serialisierungsform für HTML unter Verwendung von XML. Mit HTML5 wurde die Spezifikation nicht mehr durch die Standard Generalized Markup Language (SGML) - eine Metasprache zur Definition von Auszeichnungssprachen - sondern durch ein Dokument-Objekt-Modell (DOM) beschrieben. Die in dieser Version neu eingeführten Elemente sollten es erlauben HTML Dokumente semantisch vernünftiger zu strukturieren. (vgl. [CG12] S.20ff) Im Oktober 2014 wurde HTML5 dann vom W3C zum De-facto Standard des WWW erklärt. Heute existiert neben der Spezifikation des W3C auch noch ein sogenannter „lebender Standard“ der Web Hypertext Application Technology Working Group (WHATWG). Die WHATWG ist ein Zusammenschluss von Unternehmen wie zum Beispiel Mozilla Foundation, Opera Software und Apple. Der allgemeine Sprachgebrauch von HTML ist dadurch nicht an die W3C Spezifikation gebunden. Er erstreckt sich über den „lebenden Standard“ der WHATWG hinaus und beinhaltet zahlreiche Schnittstellen zu anderen Technologien. Abbildung 1 verdeutlicht diese Situation.

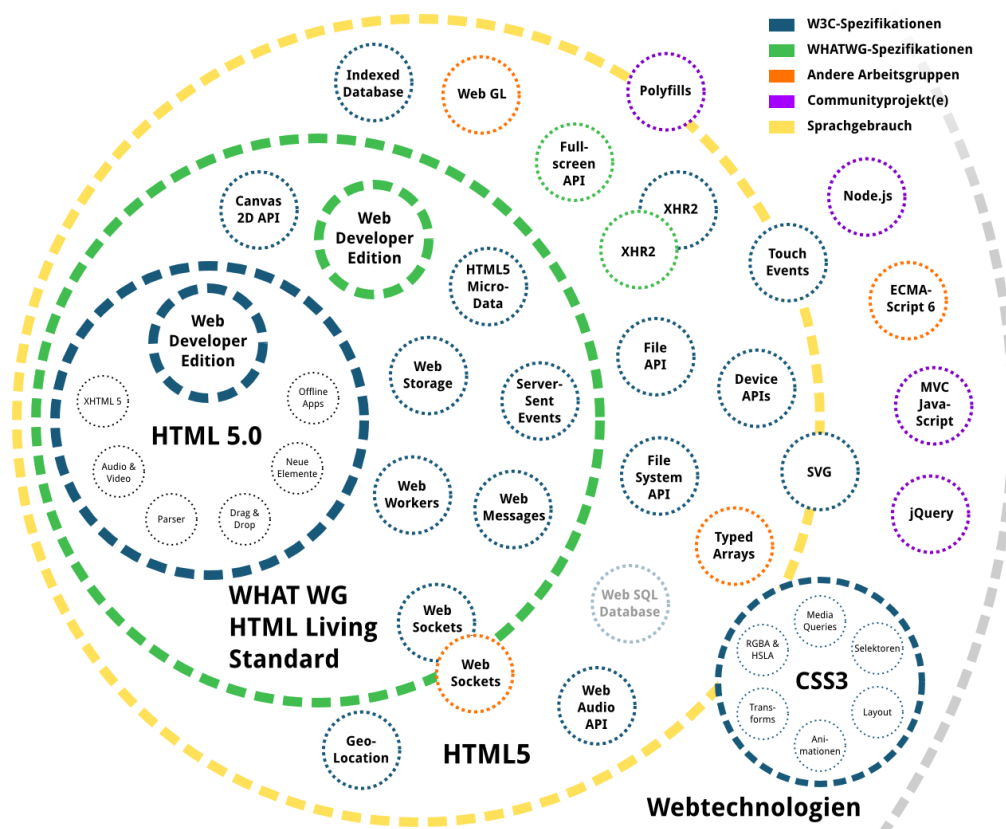


Abbildung 1: HTML5 Spezifikationen Übersicht

Ziele HTML5 wurde mit besonderem Augenmerk auf die Kompatibilität entwickelt. Vorhandene Spezifikationen wie HTML4.01, XHTML1.0 und DOM2 sollten unter einem Dach gebündelt werden. Hierdurch wird der vorangegangenen Fragmentierung entgegen gewirkt. Schon vorhandene Inhalte müssen weitestgehend unterstützt werden auch wenn sie nicht zur HTML5 Spezifikation gehören. Beispielsweise werden fehlerhaft verschachtelte Elemente trotzdem akzeptiert. *Graceful degradation* ist als ein weiteres Ziel für HTML5 definiert worden und bedeutet soviel wie „Schrittweise Abstufung“. Es stellt sicher, dass ein HTML Dokument auch dann verarbeitet wird sollte der verwendete Browser ein bestimmtes benutztes Element nicht unterstützen. Weiter galt für die Spezifikation, dass schon vorhandene Techniken, die weitläufig verbreitet sind, nicht neu entwickelt werden sollten. Stattdessen sollten sie übernommen werden. Dies beruht auf dem Umstand, dass die Browser Hersteller jeweils ihre eigenen Techniken bevorzugen und weiter entwickeln und dadurch auch für ihre Verbreitung sorgen. Evolution statt Revolution stand über den Zielen von HTML5. (X)HTML wurde weiter entwickelt und nicht von Grund auf neu definiert. So ist in Tabelle 1 und Tabelle ?? die zum aktuellen Zeitpunkt verfügbare Unterstützung von HTML5 in den gängigsten Browsern abzulesen.

| Hersteller | Desktop/Mobile | Version |
|----------------|-------------------|---------|
| Mozilla | Firefox | 4.0 |
| | Firefox Mobile | 16 |
| Google | Chrome | 10 |
| | Chrome Mobile | 25 |
| | Android | 4.0 |
| Apple | Safari | 5.1 |
| | Safari iOS | 5.1 |
| Microsoft | Internet Explorer | 10 |
| | Windows Phone | 8 |
| Opera Software | Opera | 11.64 |
| Blackberry | Browser | 10 |

Tabelle 1: HTML5 Browserkompatibilität

Aufbau Ein jedes HTML Dokument beginnt mit dem sogenannten *Doctype*. Dieser legt fest wie das Dokument vom Browser verarbeitet werden soll. Verschiedene Varianten wie *strict*, *transitional* und *frameset* sind in HTML5 nicht vorgesehen. In den Vorgängerversionen musste die Variante jedoch mit angegeben werden um eine eindeutige Interpretation des Dokuments zu gewährleisten. Listing 1 zeigt die beiden *Doctype* von HTML 4.01 und XHTML 1.0. Durch die Abwärtskompatibilität von HTML5 sind

auch diese *Doctype* heute noch gültig und das Dokument würde korrekt vom Parser interpretiert werden.

```

1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
2      "http://www.w3.org/TR/html4/strict.dtd">
3  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
4      "http://www.w3.org/TR/html4/loose.dtd">
5  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
6      "http://www.w3.org/TR/html4/frameset.dtd">
7  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
8      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

```

Listing 1: (X)HTML4.01 Doctype

Listing 2 hingegen zeigt das *Doctype* von HTML5. Es wurde enorm gekürzt im Vergleich zu dem *Doctype* von HTML 4.01 und XHTML 1.0. Groß- und Kleinschreibung ist nicht von Bedeutung innerhalb des *Doctype*.

```

1  <!DOCTYPE html>

```

Listing 2: HTML5 Doctype

Nach dem *Doctype* folgt der restliche Dokument Aufbau der HTML Syntax. Diese teilt sich auf in sogenannte *Tags* und *Attribute*, die diesen *Tags* zugeordnet und mit Werten versehen werden können. Es existieren für die meisten *Tags* Start- und End-*Tags*. Für einige *Tags* sind die Start- bzw. End-*Tags* optional und für einige wiederum verpflichtend in der Dokumentstruktur zu setzen. In Listing 3 sieht man ein valides HTML5 Dokument mit den Mindestanforderungen. (vgl. [Krö11] S.58)

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Beispiel Seite</title>
5    </head>
6    <body>
7      <h1>Beispiel Seite</h1>
8      <p>Dies ist ein <a href="demo.html">einfaches</a> Beispiel.</p>
9      <!-- Dies ist ein Kommentar -->
10   </body>
11 </html>

```

Listing 3: HTML5 Basis Dokument

Wichtige neue Sprachelemente In HTML5 wurde das Mikrodaten Application-Programming-Interface (API) mit aufgenommen. Mit Mikrodaten bietet sich eine weitere Möglichkeit ein HTML Dokument semantisch zu spezifizieren. Metadaten wie z.B. der verwendete Zeichensatz lassen sich so festlegen. Browser und Webseiten können über die Mikrodaten-API die gesetzten Werte auslesen und weiter verarbeiten. Auch Suchmaschinen haben natürlich Zugriff auf die Metadaten, verwenden sie jedoch heutzutage weitestgehend nicht. Aus diesem Grund tragen die Metadaten zwar zur semantischen Struktur des HTML bei, können aber aus Sicht der Suchmaschinenoptimierung getrost vernachlässigt werden. (vgl. [Selb]) Listing 4 zeigt Beispielhaft den Head Bereich eines HTML Dokuments mit vier Meta-*Tags*. Unter anderem wird mit dem ersten *Tag* der Zeichensatz näher definiert.

```

1 <head>
2   <meta charset="utf-8">
3   <meta name="viewport" content="width=device-width;" />
4   <meta name="keywords" content="Lorem ipsum">
5   <meta name="author" content="dolor sem it">
6 </head>

```

Listing 4: HTML5 Microdata

// Listing 5

```

1 <body>
2   <header>
3     
4     <h1>Ueberschrift </h1>
5   </header>
6
7   <footer>
8     <a href="kontakt.html">Kontakt </a>
9   </footer>
10 </body>

```

Listing 5: HTML5 Header und Footer

// Listing 6

```

1 <body>
2   <header>
3     <nav>
4       <ul>
5         <li><a href="#link_1.html">Wiki</a></li>

```

```

6         ...
7     </ul>
8 </nav>
9 </header>
10 <main>
11 <article>
12     <h1>Ueberschrift </h1>
13     <p>Dies ist eine Beispiel HTML5-Seite </p>
14 </article>
15
16 <aside>
17     <section>
18         <h2>Kontakt </h2>
19         <ul>
20             <li><a href="link_1.html">Wiki</a></li>
21             ...
22         </ul>
23     </section>
24 </aside>
25 </main>
26
27 <footer>
28 </footer>
29 </body>

```

Listing 6: HTML5 Struktur Elemente

// Listing 7

```

1 <head>
2     <meta charset="utf-8">
3     <!--[if lt IE 9]>
4         <script src="//html5shim.googlecode.com/svn/trunk/html5.js"></
          script>
5     <![endif]-->
6     <title>HTML5-Seite mit Grundstruktur</title>
7 </head>

```

Listing 7: HTML5 Internet Explorer Fallback

// Listing 8

```

1 <body>
2     <input type="email" name="addr">
3     <input type="date" name="start">

```

```
4  <input type="time"   min="09:00" max="17:00" >
5  <input type="date"   required="required" name="start">
6  </body>
```

Listing 8: HTML5 Formular Eingabe

2D Canvas (canvas tag), Media (audio/video tags), geänderte Elemente (b, i, hr, small, ...)

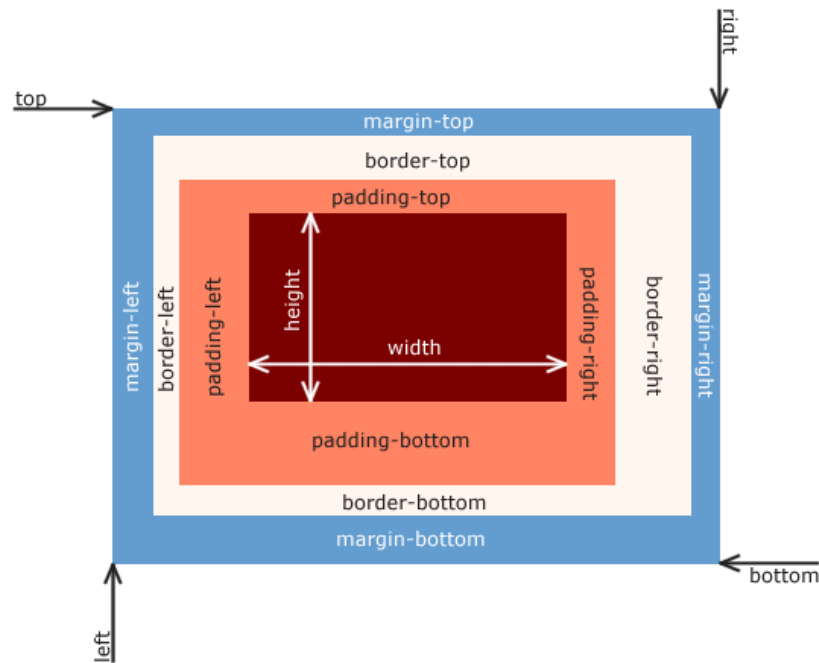
2.2 CSS3

// Allgemeiner Aufbau - Gestaltungssprache, Kern Element des WWW, Darstellung und Inhalt getrennt, Unterschiedliche Optik je nach Ausgabe Gerät
// Syntax - Selektoren, Eigenschaften, Werte, Pseudoklassen
// Listing 9

```
1  Selektor [, Selektor2 , ...]
2  {
3      Eigenschaft1 : Wert1;
4      ...
5      EigenschaftN : WertN[;]
6  }
7  /* Kommentar – In eckigen Klammern stehen optionale Angaben */
```

Listing 9: CSS3 Syntax Beispiel

// CSS-Box-Modell - margin, border, padding
// Abbildung 2

Abbildung 2: CSS-Boxmodell¹

// Medienspezifische Stylesheets (@media print, screen, ...)
 // Listing 10

```

1  @media print {
2    body {
3      color: black;
4      background-color: white;
5    }
6    .navigation {
7      display: none;
8    }
9  }
```

Listing 10: CSS3 medienspezifisches Stylesheet

// Eigenschaftsspezifische Stylesheets (@media screen and (max-width:1024px))
 // Listing 11

```

1  #inhalt {
2    width: 800px;
3  }
4
5  @media screen and (max-width: 1024px) {
```

¹Quelle: <http://commons.wikimedia.org/wiki/File:Boxmodell-detail.png>

```

6  #inhalt {
7      width: 600px;
8  }
9
10  aside {
11      display: none;
12  }
13  }

```

Listing 11: CSS3 eigenschaftsspezifisches Stylesheet

```
// Verzahnung mit HTML5 - link tag, style tag, html tag, @import innerhalb Stylesheet
// Listing 12 - Einbindung über Link Tag
```

```
1  <link rel="stylesheet" type="text/css" href="beispiel.css" />
```

Listing 12: Stylesheet Einbindung über Link Tag

```
// Listing 13 - Einbindung über Style Tag
```

```

1  <head>
2      <title>Dokument mit Formatierungen</title>
3      <style type="text/css">
4          body { color: purple; background-color: #d8da3d; }
5      </style>
6  </head>

```

Listing 13: Stylesheet Einbindung über Style Tag

```
// Listing 14 - Einbindung in HTML Tag
```

```
1  <span style="font-size: small;">Text</span>
```

Listing 14: Stylesheet Einbindung in HTML Tag

2.3 JavaScript

```
// Grundlagen - Geschichte, Sicherheit, aktueller Stand
// Listing 15 - Einbindung als separate Datei im Head
```

```
1  <script src="script.js" type="text/javascript"></script>
```

Listing 15: JavaScript Einbindung als separate Datei im Head

// Listing 16 - Einbindung in Skript Tag im Head und Body

```
1 <script type="text/javascript"></script>
```

Listing 16: JavaScript Einbindung in Skript Tag im Head und Body

// Sprachelemente - Kommentare, Funktionen, Objekte

// Variablen - Dynamische Typisierung(Loose Typing), Case-Sensitive, ungarische Nomenklatur, spezielle Werte(*undefined*, *null*, *true*, *false*, *NaN*)

// Operatoren - *+*, *-*, ***, */*, zusätzlich *+* als Zeichenverkettung, In- und Dekrement, Zuweisung, Vergleich, *typeof*, Logisch

// Kontrollstrukturen - *if*, *switch*, *for*, *while* Anweisungen inklusive ihrer Varianten

// Document Object Model - Schnittstelle zum HTML Aufbau, W3C Spezifikation unterschiedlich implementiert, Knoten Beziehungen, Verarbeitung des DOM, Generierung von HTML durch Serialisierung, Listing 17 beschreiben und zur Baumstruktur hinleiten

// Listing 17

```
1 <table>
2   <thead>
3     <tr>
4       <th>Vorname</th>
5       <th>Name</th>
6     </tr>
7   </thead>
8   <tbody>
9     <tr>
10      <td>Donald</td>
11      <td>Duck</td>
12    </tr>
13  </tbody>
14 </table>
```

Listing 17: HTML5 Beispiel Definition

// Abbildung 3

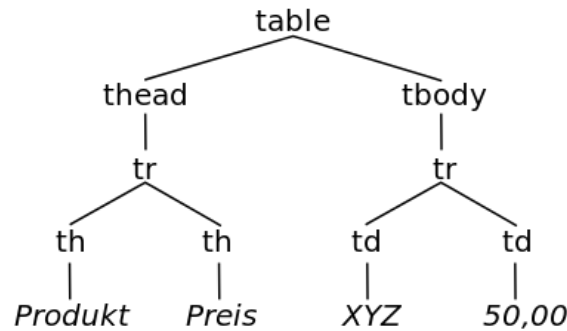


Abbildung 3: DOM Beispielsbaum

// Ereignisse

// Übersicht einiger wichtiger Events

- onabort (bei Abbruch)
- onblur (beim Verlassen)
- onchange (bei erfolgter Änderung)
- onclick (beim Anklicken)
- ondblclick (bei doppeltem Anklicken)
- onerror (im Fehlerfall)
- onfocus (beim Aktivieren)
- onkeydown (bei gedrückter Taste)
- onkeypress (bei gedrückt gehaltener Taste)
- onkeyup (bei losgelassener Taste)
- onload (beim Laden einer Datei)
- onmousedown (bei gedrückter Maustaste)
- onmousemove (bei weiterbewegter Maus)
- onmouseout (beim Verlassen des Elements mit der Maus)
- onmouseover (beim Überfahren des Elements mit der Maus)
- onmouseup (bei losgelassener Maustaste)
- onreset (beim Zurücksetzen des Formulars)
- onselect (beim Selektieren von Text)
- onsubmit (beim Absenden des Formulars)
- onunload (beim Verlassen der Datei)

jQuery

// jQuery Bibliothek beinhaltet Elementselektion, Funktionen zum DOM, Animationen und Effekte, AJAX Funktionalitäten

// Selektoren

// Ereignisse - unterschiede zum JS Standard bei der Definierung, Einfachheit

// Übersicht der wichtigsten Funktionen zu Events

- .bind – Handler an Event binden
- .on – Handler an Event binden
- .blur – Ereignis, wenn ein Element den Fokus verliert
- .click – Klick mit der Maustaste
- .dblclick – Doppelklick mit der Maustaste
- .hover – Mauszeiger bewegt sich über ein Element
- .mousemove – Mauszeiger bewegt sich in einem Element
- .keypress – eine Taste der Tastatur wird gedrückt
- .keyup – eine Taste der Tastatur wird losgelassen
- .change – ein Formularfeld wird verändert

// DOM-Manipulation

// AJAX

2.4 ABAP

// Herkunft/Entstehung

// Grundlagen

// Wichtige Elemente (OpenSQL)

2.5 SAP UI5 Framework

2.5.1 Definition

// Aufbauend auf jQuery, AJAX, HTML5/CSS3 [Ant14]

2.5.2 Architektur

// Einführung in SAPUI5 S. 123 [Ant14]

// Abbildung 4

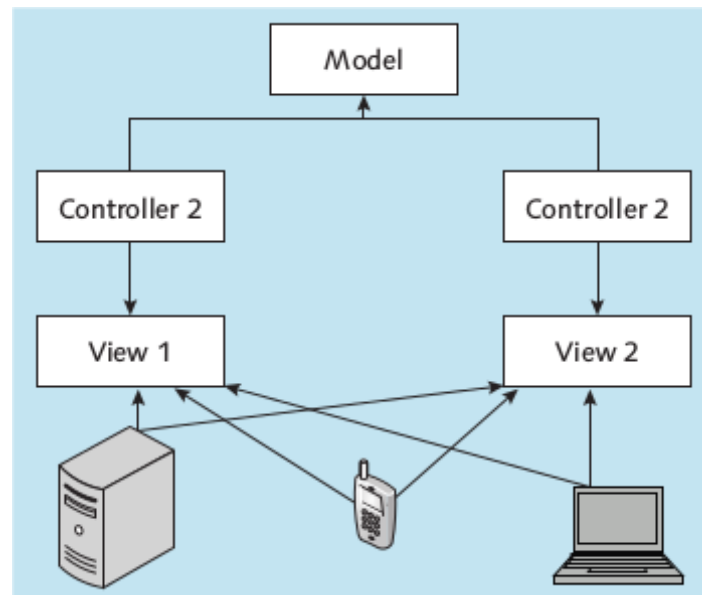


Abbildung 4: Model-View-Controller-Architekturmuster [Ant14]

2.5.3 OData Protokoll

// Einführung in SAPUI5 S. 168

// SAP Netweaver Gateway OData Services

3 Software Ergonomie

// Beleg für die Wichtigkeit von Software Ergonomie
// Kurze Übersicht über das Themenfeld Software Ergonomie
// Wichtigsten Aspekte nennen und näher erläutern

3.1 Definition

Kognitionspsychologie // Modellierung und Simulation von menschlichen Denk- und Wahrnehmungsprozessen

Arbeitsphysiologie, Industrieanthropologie // Beschäftigung mit grundlegenden menschlichen Fähigkeiten zur Informationsaufnahme und Informationsverarbeitung

Arbeitspsychologie // Untersuchung der Wechselbeziehungen zwischen Arbeit, deren Schnittstellen und psychischen Faktoren (unter anderem Arbeitszufriedenheit und -unlust)

3.2 DIN EN ISO 9241

// DIN Norm zur Software Ergonomie
// Die 7 Grundsätze der Dialoggestaltung:

- Aufgabenangemessenheit
- Selbstbeschreibungsfähigkeit
- Erwartungskonformität
- Fehlertoleranz
- Steuerbarkeit
- Individualisierbarkeit
- Lernförderlichkeit

DIN EN ISO 14915

// Erweiterung der ISO 9241

3.3 Analyse Methoden

Eye Tracking // Funktionsweise und Ergebnis

Mouse Clicking // Funktionsweise und Ergebnis

3.4 SAP Technologien in Bezug auf Software Ergonomie

3.4.1 Business Server Pages

// Business Server Pages (BSP) ist old school Technik

// geklaut von Java Server Pages (JSP)

3.4.2 Web Dynpro for ABAP

// Aktuelle Technik

// ABAP Code generiert HTML

// statischer und dynamischer Teil

3.4.3 SAP Fiori / SAP UI5 / SAP Screen Personas

// cutting edge

// aktuelle SAP UI Strategie

// SAP Präsi Chart Fiori/SP renew, etc. pp

// SAP Fiori einerseits Name des Themes/Guideline

// andererseits Bündel der gängigsten TAs/GPs als fertige

// Mobile First/Responsive Design Applikationen

// SAP UI5 - SAPs Framework zur Entwicklung von eigenen Applikationen im Fiori Style

// Nicht zu tief auf JS, HTML etc eingehen, dass kommt im nächsten Kapitel

// SAP SP - Zusätzliche Schicht um Standard Dynpro zu Personalisieren und so

4 Fallbeispiel SAP UI5

Lorem ipsum dolor sit amet.

4.1 Beschreibung

// Frontend - Browser, Elemente
// Backend - JSON, OData Model
// Analyse der wichtigen Arbeitsschritte

4.2 Hilfsmittel

4.2.1 Entwicklungsumgebung

// Kurze Beschreibung der Entwicklungsumgebung
// Sprich Eclipse, SE80, Chrome Dev-tools, Debugger
// Neptune Application Designer

4.2.2 UI Design und Prototyping

// Wireframing als Prototyping
// Abbildung Wireframesketcher

4.3 Implementierung

4.3.1 View

// Auszugsweise Coding bringen um bestimmte Elemente aus der Theorie zu zeigen
// Generellen Aufbau der Views erklären
// Kapselung wird dadurch verdeutlicht
// Listing 18

```
1  sap.ui.jsview("abat.Mockup.view.App", {  
2  
3    getControllerName: function () {  
4      return "abat.Mockup.view.App";  
5    },  
6  
7    createContent: function (oController) {  
8      // to avoid scroll bars on desktop  
9      this.setDisplayBlock(true);
```



```

10
11      // create app
12      this.app = new sap.m.SplitApp();
13
14      // load the master page
15      var master = sap.ui.xmlview("Master", "abat.Mockup.view.Master");
16      master.getController().nav = this.getController();
17      this.app.addPage(master, true);
18
19      // load the empty page
20      var empty = sap.ui.xmlview("Empty", "abat.Mockup.view.Empty");
21      this.app.addPage(empty, false);
22
23      // wrap app with shell
24      return new sap.m.Shell("Shell", {
25          title : "{i18n>ShellTitle}",
26          showLogout : false,
27          app : this.app
28      });
29  }
30  });

```

Listing 18: Root View der Applikation

```

// Master/Detail Applikation mit Fragment und Chart View Aufbau
// TODO: Visio Diagramm oder vergleichbares erstellen
// sap.ui.view
// --- sap.m.Shell
// --- sap.m.SplitApp
// --- sap.m.Page
// --- sap.m.Bar
// --- sap.m.Bar
// --- sap.m.List
// --- sap.m.ObjectListItem
// --- sap.m.Bar
// --- sap.m.Page
// --- sap.m.ObjectHeader
// --- sap.m.ObjectAttribute
// --- sap.m.ObjectAttribute
// --- sap.m.ObjectAttribute
// --- sap.m.ObjectAttribute
// --- sap.m.ObjectStatus

```

```
// — — — —- sap.IconTabBar
// — — — —- sap.IconTabFilter
// — — — —- sap.ui.core.Fragment
// — — — —- sap.ui.core.FragmentDefinition
// — — — —- sap.viz.ui5.Bar
// — — — —- sap.IconTabFilter
// — — — —- sap.ui.core.Fragment
// — — — —- sap.ui.core.FragmentDefinition
// — — — —- sap.viz.ui5.Bar
// — — — —- sap.m.Bar
```

4.3.2 Model und Controller

```
// die Verbindung von beiden Anhand von Coding zeigen
// TODO: OData Modell einbinden
// Listing 19
```

```
1  ...
2  // JSON Modell an die Root View binden
3  var oModel = new sap.ui.model.json.JSONModel("model/mock.json");
4  oView.setModel(oModel);
5
6  // OData Modell
7  var oModel = new sap.ui.model.odata.ODataModel(<URL>);
8  oView.setModel(oModel);
9
10 // I18N(Lokalisierung) Modell
11 var i18nModel = new sap.ui.model.resource.ResourceModel({
12     bundleUrl : "i18n/messageBundle.properties"
13 });
14 oView.setModel(i18nModel, "i18n");
15
16 // Geraetespezifisches Modell
17 var deviceModel = new sap.ui.model.json.JSONModel({
18     isPhone : jQuery.device.is.phone,
19     listMode : (jQuery.device.is.phone) ? "None" : "SingleSelectMaster",
20     listItemType : (jQuery.device.is.phone) ? "Active" : "Inactive"
21 });
22 deviceModel.setDefaultBindingMode("OneWay");
23 oView.setModel(deviceModel, "device");
24 ...
```

Listing 19: Component.js - Datenmodell an die Root View binden

4.3.3 Backend

```
// ABAP Stack der den RESTful Service bereitstellt zeigen  
// Beispielhafte Implementation des HTTP Responses
```

5 Analyse

5.1 Heatmap

// Angewandte Analyse mit Heatmap

5.2 UI-Objekte

// Mobile First/Responsive Design

5.3 PLATZHALTER

// PLATZHALTER

6 Schluss

Lorem ipsum dolor sit amet.

Zusammenfassung // Arbeitsgebiete, Produktions & Dienstleistungsbereiche
// Arbeitsergebnisse
// Projektziele, Projektergebnisse, Projekttermine
// Mitwirkungszeiträume
// Liste aller selbst wahrgenommen Aufgaben und Tätigkeiten
// Projektmeilensteine
// Ablauforganisation & Beteiligte
// Arbeitsformen, Arbeitsmittel, Arbeitsabläufe
// Kommunikations- / Informationsgewohnheiten
// Auswertung relevanter Literatur
// Themen aus Lehrveranstaltungen

Bewertung // Wesentliche Erkenntnisse und Erfahrungen
// Folgerungen und Konsequenzen
// Vorschläge für Verbesserung und Veränderung
// Auswirkungen auf persönliche Berufs- und Karriereplanung
// Bezug zum Studium
// hilfreiche Studieninhalte
// neu gewonnenes Interesse

7 Quellenverzeichnis

- [Ant14] ANTOLOVIC, Miroslav: *Einführung in SAPUI5: [Einführung in das SAP UI Development Toolkit für HTML5 ; moderne Benutzeroberflächen gestalten und erweitern ; Programmiermodell, Controls und UI-Elemente in der Praxis einsetzen]*. 1. Aufl. Bonn [u.a.] : Galileo Press, 2014 (SAP PRESS). – ISBN 9783836227537 und 3836227533
- [CG12] CLEMENS GULL, Stefan M.: *HTML5-Handbuch: [die neuen Features von HTML5 ; Webseiten für jedes Endgerät: Media Queries für mobile Devices ; so setzen Sie anspruchsvolle Web-Layouts mit HTML5 und CSS um ; umfangreicher Referenzteil für HTML und CSS zum Nachschlagen ; zukunftsorientierte Webseiten erstellen]*. 2., aktualisierte und erw. Aufl. Haar bei München : Franzis, 2012 (Know-how ist blau). – ISBN 3645601511 und 9783645601511
- [Krö11] KRÖNER, Peter: *HTML5: Webseiten innovativ und zukunftssicher ; [neu: Web Workers, File API, IE 9 uvm.]*. 2. Aufl. München : Open Source Press, 2011 (Professional reference). – ISBN 3941841343 und 9783941841345
- [Sela] SELFHTML: *Physische Auszeichnungen im Text*. <http://de.selfhtml.org/html/text/physisch.htm>. – Zugriff: 01.12.2014, Archiviert mit WebCite®: <http://www.webcitation.org/6UUqSW9ao>
- [Selb] SELFHTML: *Suchmaschinenoptimierung*. <http://wiki.selfhtml.org/wiki/Suchmaschinenoptimierung>. – Zugriff: 04.12.2014, Archiviert mit WebCite®: <http://www.webcitation.org/6UZA8MTZe>

Anhang

A GUI

Ein toller Anhang.

Screenshot

Unterkategorie, die nicht im Inhaltsverzeichnis auftaucht.

Erklärung

Hiermit versichere ich, dass ich meine Abschlussarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Datum:

.....

(Unterschrift)