

Webkomponenten zur Entwicklung von Serveranwendungen

Andreas Hahn



BACHELORARBEIT

Nr. S1510238020-A

eingereicht am
Fachhochschul-Bachelorstudiengang

Medientechnik und -design

in Hagenberg

im Februar 2018

Diese Arbeit entstand im Rahmen des Gegenstands

Bachelorarbeit 1

im

Sommersemester 2017

Betreuung:

Rimbert Rudisch-Sommer

Erklärung

Ich erkläre eidesstattlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen entnommenen Stellen als solche gekennzeichnet habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Hagenberg, am 28. Februar 2018

Andreas Hahn

Inhaltsverzeichnis

Erklärung	iii
Vorwort	vi
Kurzfassung	viii
Abstract	ix
1 Motivation	1
1.1 Einleitung	1
1.2 Theoretischer Hintergrund und Stand der Forschung	1
1.3 Forschungsfrage	2
1.4 Methodik	2
1.5 Erwartete Ergebnisse	2
2 State of the Art	3
2.1 Verwendung von Komponenten im Web	3
2.2 Webkomponenten	3
2.3 Komponentenorientierte Frameworks	4
3 Grundlegendes Konzept	5
3.1 Anforderung an die komponentenorientierte Webanwendung	5
3.2 Eigene Idee, technisches Design	5
4 Implementierung der Webanwendung	6
4.1 Struktur der Webanwendung	6
4.2 Umsetzung mit Dependencies Electron / ScramEngine	6
4.3 Implementierung der Komponenten	6
5 Evaluierung	7
5.1 Aufwand / Nutzen	7
5.2 Wiederverwendbarkeit	7
5.3 Skalierbarkeit, Flexibilität	7
5.4 Testbarkeit	7
6 Zusammenfassung	8
6.1 Kosten-Nutzen Faktor	8

Inhaltsverzeichnis	v
Quellenverzeichnis	9
Literatur	9

Vorwort

Dies ist **Version 2017/02/28** der LaTeX-Dokumentenvorlage für verschiedene Abschlussarbeiten an der Fakultät für Informatik, Kommunikation und Medien der FH Oberösterreich in Hagenberg, die mittlerweile auch an anderen Hochschulen im In- und Ausland gerne verwendet wird.

Das Dokument entstand ursprünglich auf Anfragen von Studierenden, nachdem im Studienjahr 2000/01 erstmals ein offizieller LaTeX-Grundkurs im Studiengang Medientechnik und -design an der FH Hagenberg angeboten wurde. Eigentlich war die Idee, die bereits bestehende *Word*-Vorlage für Diplomarbeiten „einfach“ in LaTeX zu übersetzen und dazu eventuell einige spezielle Ergänzungen einzubauen. Das erwies sich rasch als wenig zielführend, da LaTeX, vor allem was den Umgang mit Literatur und Grafiken anbelangt, doch eine wesentlich andere Arbeitsweise verlangt. Das Ergebnis ist – von Grund auf neu geschrieben und wesentlich umfangreicher als das vorherige Dokument – letztendlich eine Anleitung für das Schreiben mit LaTeX, ergänzt mit einigen speziellen (mittlerweile entfernten) Hinweisen für *Word*-Benutzer. Technische Details zur aktuellen Version finden sich in Anhang ??.

Während dieses Dokument anfangs ausschließlich für die Erstellung von Diplomarbeiten gedacht war, sind nunmehr auch *Masterarbeiten*, *Bachelorarbeiten* und *Praktikumsberichte* abgedeckt, wobei die Unterschiede bewusst gering gehalten wurden.

Bei der Zusammenstellung dieser Vorlage wurde versucht, mit der Basisfunktionalität von LaTeX das Auslangen zu finden und – soweit möglich – auf zusätzliche Pakete zu verzichten. Das ist nur zum Teil gelungen; tatsächlich ist eine Reihe von ergänzenden „Paketen“ notwendig, wobei jedoch nur auf gängige Erweiterungen zurückgegriffen wurde. Selbstverständlich gibt es darüber hinaus eine Vielzahl weiterer Pakete, die für weitere Verbesserungen und Feinessen nützlich sein können. Damit kann sich aber jeder selbst beschäftigen, sobald das notwendige Selbstvertrauen und genügend Zeit zum Experimentieren vorhanden sind. Eine Vielzahl von Details und Tricks sind zwar in diesem Dokument nicht explizit angeführt, können aber im zugehörigen Quelltext jederzeit ausgeforscht werden.

Zahlreiche KollegInnen haben durch sorgfältiges Korrekturlesen und konstruktive Verbesserungsvorschläge wertvolle Unterstützung geliefert. Speziell bedanken möchte ich mich bei Heinz Dobler für die konsequente Verbesserung meines „Computer Slangs“, bei Elisabeth Mitterbauer für das bewährte orthographische Auge und bei Wolfgang Hochleitner für die Tests unter Mac OS.

Die Verwendung dieser Vorlage ist jedermann freigestellt und an keinerlei Erwähnung gebunden. Allerdings – wer sie als Grundlage seiner eigenen Arbeit verwenden möchte, sollte nicht einfach („ung’schaut“) darauf los werken, sondern zumindest die wichtigsten

Teile des Dokuments *lesen* und nach Möglichkeit auch beherzigen. Die Erfahrung zeigt, dass dies die Qualität der Ergebnisse deutlich zu steigern vermag.

Der Quelltext zu diesem Dokument sowie das zugehörige LaTeX-Paket sind in der jeweils aktuellen Version online verfügbar unter

<https://github.com/Digital-Media/HagenbergThesis>.

Trotz großer Mühe enthält dieses Dokument zweifellos Fehler und Unzulänglichkeiten – Kommentare, Verbesserungsvorschläge und passende Ergänzungen sind daher stets willkommen, am einfachsten per E-Mail direkt an mich:

Dr. Wilhelm Burger, Department für Digitale Medien,
Fachhochschule Oberösterreich, Campus Hagenberg (Österreich)
wilhelm.burger@fh-hagenberg.at

Übrigens, hier im Vorwort (das bei Diplom- und Masterarbeiten üblich, bei Bachelorarbeiten aber entbehrlich ist) kann kurz auf die Entstehung des Dokuments eingegangen werden. Hier ist auch der Platz für allfällige Danksagungen (z. B. an den Betreuer, den Begutachter, die Familie, den Hund, ...), Widmungen und philosophische Anmerkungen. Das sollte allerdings auch nicht übertrieben werden und sich auf einen Umfang von maximal zwei Seiten beschränken.

Kurzfassung

Webkomponenten tragen zur Atomisierung der Webprogrammierung bei, da diese den WebentwicklerInnen die Möglichkeiten bieten, bestehende HTML Elemente zu erweitern und eigene zu kreieren. Dieser unscheinbarer Zusatz ermöglicht ProgrammiererInnen ihre Software aus mächtigen, zusammengesetzten Komponenten zu erstellen. Diese Bachelorarbeit konzipiert ein Gerüst aus Webkomponenten in Kombination mit anderen gängigen Webtechnologien, welche die Erstellung einer Serveranwendung weitgehend mit Webkomponenten ermöglicht und testet dieses anhand eines im fünften Semester entwickelten Prototypen.

Abstract

Webcomponents aim towards atomarizing web programming by giving developers the possibility to extend already existng HTML elemnts and to create own components. This modest addition enables programmers to build their software out of combined, powerfull components. This bachelor thesis focuses on the development of a framework of web components in combination with other popular web technologies to develop a server application using webcomponents. This framework will then be evaluated based on a prototype implemented as a project in the fifth semester.

Kapitel 1

Motivation

1.1 Einleitung

Die Idee, die Webprogrammierung komponentenorientiert zu gestalten, gibt es seit 1998 [4]. Jedoch hat sich keine dieser Ideen bis heute durchgesetzt. Seit einigen Jahren gibt es dazu wieder Ansätze, welche von größeren Firmen, wie beispielsweise Google mit Angular¹ oder Facebook mit React², mittels eigen entwickelten komponentenbasierten Frameworks versuchen, die Webentwicklung voran zu treiben. Allerdings ist seit neuestem die grundlegende Idee der nativen Webkomponenten³, welche künftig nur als Webkomponenten bezeichnet werden, zurück. Da diese, sobald gängige Webbrowser alle Technologien von Webkomponenten – HTML Imports, Templates, Custom Elements, Shadow DOM – unterstützen, einen weiteren Schritt in standardisierte, wiederverwendbare und unabhängige Webbausteine bedeutet. Zur Zeit ist es noch nicht, oder erschwert möglich, clientseitig und serverseitig auf Webkomponenten zu setzen. Diese Bachelorarbeit beleuchtet jene Problematik und stellt Lösungsvorschläge anhand eines konkreten Content-Management-System dar.

1.2 Theoretischer Hintergrund und Stand der Forschung

Ein Content-Management-System (CMS) ist ein System, welches den NutzerInnen entsprechende Inhalte verwalten lässt. Seien es beispielsweise die Kunden, die Projekte oder aber interne Statistiken einer Firma. BenutzerInnen mit Zugriffsrechten können dieses System meist ohne Programmier- oder HTML-Kenntnisse bedienen. Angular oder React wären für die Umsetzung eines solchen CMS geeignet. Aus heutiger Sicht zählen diese Technologien zu den gängigsten Frameworks und werden von zahlreichen EntwicklerInnen stetig verbessert. Diese besagten Frameworks sind mit einer steilen Lernkurve behaftet und benötigen Zeit um sich damit vertraut zu machen.

Im Gegensatz dazu sind Webkomponenten logisch deklariert und bauen lediglich auf Kenntnissen von nativem HTML, CSS und Javascript auf. Die Unterstützung wird noch nicht von jedem Browser gewährleistet, was einige EntwicklerInnen noch zögern lässt

¹<https://angular.io/>

²<https://facebook.github.io/react/>

³<https://www.webcomponents.org/>

um komplett auf Webkomponenten zu setzen. Google hat mit Polymer⁴ eine Javascript-Bibliothek geschaffen, welche die Kreierung und Nutzung von Webkomponenten ungemein vereinfacht. Darüber hinaus nutzt diese Bibliothek Polyfills, was die Funktionen, welche Polymer benötigt, bei Bedarf liefert und wird dadurch nicht nur in allen gängigen Webbrowsern – Chrome, Firefox, Edge, Safari – sondern auch in deren älteren Versionen, die die notwendigen Funktionen nicht implementiert haben, unterstützt [3]. Native Browserunterstützung ist bereits bei den meisten in Entwicklung und würde Polyfills überflüssig machen.

1.3 Forschungsfrage

Aus diesen Ansätzen ergibt sich die folgende Forschungsfrage für diese Bachelorarbeit:

Wie muss ein CMS programmatisch konzipiert werden, um die grundlegenden Funktionen weitgehend mittels server- und clientseitigen Webkomponenten zu realisieren.

1.4 Methodik

Um diese Frage zu beantworten, soll die Bachelorarbeit als eine Kombination von Literaturarbeit und praktischer bzw. prototypischer Umsetzung realisiert werden.

Zunächst soll aus bestehender Literatur (erweiternd zu Abschnitt 1.2) erörtert werden, wie mit dem Thema der Webkomponenten umgegangen wird, und wie mit komponentenorientierte Frameworks. Gemeinsame Faktoren wie Mechaniken sollen daraus extrahiert werden und als Grundlage für ein eigenes, theoretisches Konzept dienen. Die Vorteile und Limitierungen von Webkomponenten werden dabei erörtert. Dieses Konzept soll schlussendlich eine Liste von Kerntechniken enthalten, welche eine Umsetzung eines CMS mittels Webkomponenten ermöglicht.

Überprüft soll die Anwendbarkeit dieses Konzept durch einen eigenen, im Rahmen des fünften Semesters entwickelten, Prototypen werden.

1.5 Erwartete Ergebnisse

Als konkretes Ergebnis wird ein Gerüst mit Webkomponenten erstellt, welches als Grundlage für die Erstellung eines CMS dienen soll. Es wird erwartet, dass sich solche konkreten Techniken finden und umsetzen lassen.

Bei den Tests der praktischen Umsetzung der Webkomponenten in einem CMS wird ebenfalls eine positive Evaluierung erwartet, da es bereits erfolgreiche Konzepte basierend auf server- und clientseitigen Webkomponenten gibt, auf deren Erfahrungen aufgebaut werden kann.

⁴<https://www.polymer-project.org/>

Kapitel 2

State of the Art

2.1 Verwendung von Komponenten im Web

Das Web besteht aus Blöcken, sogenannten Komponenten. Wenn EntwicklerInnen beispielsweise einen Link mittels einem a-Tag nutzen, wird erwartet, dass dieser sich wie ein Link-Element verhält. Dieses Element hat seine eigene standardmäßige blaue Farbe, einen Handzeiger bei hover und vor allem die Funktionalität, dass bei einem Klick auf das Element zu einer neuen URL weitergeleitet wird. Dieses Verhalten und Styling wird ohne jegliches einwirken des/der EntwicklerIn bereitgestellt. Jedes HTML Element funktioniert nach diesem Prinzip, welches HTML Code einfacher zu schreiben, aber auch verständlicher macht. Durch Kombination solcher Elemente, mit eigens definierten Stylesheets können komplexere Gerüste aus HTML Elementen, mit komplexeren CSS/Javascript entstehen. Damit nicht ein/eine EntwicklerIn die komplette Struktur im Kopf behalten muss, und bei Problemen nur diese/r das aufgetretene Problem beheben kann, hat sich bewehrt, Code in kleine, überschaubare Teile herunter zu brechen. In Einheiten, welche das gesamte Gerüst wartbar machen. Diese Einheiten können einfachere Funktionen, Module, Komponenten oder aber viele andere Ansätze sein, welche komplexe Strukturen in atomare Einzelteile aufteilen.

2.2 Webkomponenten

Webkomponenten bestehen aus 4 Bestandteilen:

- Custom Elements - APIs um neue HTML Elemente zu definieren
- HTML Imports - deklarative Methode um HTML Dokumente in andere zu importieren
- HTML Templates - `<template>` Element, welches Dokumenten eigenen DOM erlaubt
- Shadow DOM - DOM und Styling abkapseln

An einer Standardisierung der Webkomponenten wird bereits Seiten der W3C gearbeitet. Mehr dazu findet sich in [2]. Jedoch um diese derzeit in allen größeren Browsern zu nutzen gibt es sogenannte Polyfills [1]. Dieses Polyfill ist ein Codestück, welches Technologie zur Verfügung stellt, die der Browser nicht nativ unterstützt. Wie der Name –

Webkomponente – suggeriert, ermöglicht diese Technologie die Webentwicklung in kleinere, wiederverwendbare, modulare Container zu „komponentisieren“. Der klare Vorteil besteht darin, dass diese Komponenten unabhängig von einem Framework, vollständig mit „vanilla“ HTML, CSS und Javascript kreiert werden können und wie in Kap. 2.1 erwähnt, Code wartbar machen.

2.3 Komponentenorientierte Frameworks

Kapitel 3

Grundlegendes Konzept

3.1 Anforderung an die komponentenorientierte Webanwendung

3.2 Eigene Idee, technisches Design

Kapitel 4

Implementierung der Webanwendung

4.1 Struktur der Webanwendung

4.2 Umsetzung mit Dependencies Electron / ScramEngine

4.3 Implementierung der Komponenten

Kapitel 5

Evaluierung

- 5.1 Aufwand / Nutzen
- 5.2 Wiederverwendbarkeit
- 5.3 Skalierbarkeit, Flexibilität
- 5.4 Testbarkeit

Kapitel 6

Zusammenfassung

6.1 Kosten-Nutzen Faktor

Quellenverzeichnis

Literatur

- [1] Okt. 2010. URL: <https://remysharp.com/2010/10/08/what-is-a-polyfill/> (siehe S. 3).
- [2] Nov. 2017. URL: https://www.w3.org/standards/techs/components#w3c_all (siehe S. 3).
- [3] Polymer Authors. *Browser compatibility*. Google, Juli 2017. URL: <https://www.polymer-project.org/1.0/docs/browsers> (siehe S. 2).
- [4] Travis Leithead und Arron Eicholz. *Bringing componentization to the web: An overview of Web Components*. Microsoft Edge Team, Juli 2017. URL: <https://blogs.windows.com/msedgedev/2015/07/14/bringing-componentization-to-the-web-an-overview-of-web-components/#H5ykveDs14lpvXEr.97> (siehe S. 1).