



Diplomarbeit

Höhere Technische Bundeslehranstalt Leonding
Abteilung für Medientechnik

Erweiterung des WYSIWYG Texteditors "Trix" für Barrierefreiheit

Eingereicht von: **Halil Bahar, 5AHITM**

Sonja Cao, 5AHITM

Datum: **5. April 2021**

Betreuer: **Prof. Mag. Dr. Thomas Stütz**

Projektpartner: **Fabasoft Software GmbH**

Declaration of Academic Honesty

Hereby, I declare that I have composed the presented paper independently on my own and without any other resources than the ones indicated. All thoughts taken directly or indirectly from external sources are properly denoted as such.

This paper has neither been previously submitted to another authority nor has it been published yet.

Leonding, April 5, 2021

Halil Bahar, Sonja Cao

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorgelegte Diplomarbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Gedanken, die aus fremden Quellen direkt oder indirekt übernommen wurden, sind als solche gekennzeichnet.

Die Arbeit wurde bisher in gleicher oder ähnlicher Weise keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Leonding, am 5. April 2021

Halil Bahar, Sonja Cao

Gender Declaration

In this diploma thesis the language form of the generic masculine is used for reasons of better readability. All personal designations are therefore to be understood as gender-neutral.

Gender Erklärung

Aus Gründen der besseren Lesbarkeit wird in dieser Diplomarbeit die Sprachform des generischen Maskulinums verwendet. Alle personenbezogenen Bezeichnungen sind somit geschlechtsneutral zu verstehen.

Abstract

In the course of the diploma thesis of Halil Bahar and Sonja Cao an extension for the open source What You See Is What You Get (WYSIWYG) text editor Trix from the company Basecamp is being developed for the client Fabasoft Software GmbH. The previous text editor of the Fabasoft Cloud no longer meets the desired requirements and Trix is still unsuitable in terms of accessibility for people with disabilities. The extension is therefore intended to be used in products of Fabasoft and to meet the criteria of WCAG 2.1, making it accessible with screen readers and the keyboard to enable all users to write and format texts without barriers.

Zusammenfassung

Im Zuge der Diplomarbeit von Halil Bahar und Sonja Cao wird der Open Source What You See Is What You Get (WYSIWYG) Texteditor namens Trix vom Unternehmen Basecamp für den Auftraggeber Fabasoft Software GmbH erweitert. Der bisherige Texteditor der Fabasoft Cloud entspricht nicht mehr den gewünschten Anforderungen und Trix ist hinsichtlich der Accessibility für Menschen mit Beeinträchtigungen noch ungeeignet. Die Erweiterung soll daher einen Einsatz in Fabasoft Produkten ermöglichen sowie die Kriterien der WCAG 2.1 erfüllen und so mit Screenreadern sowie Tastatur zugänglich sein, um allen Benutzern die Möglichkeit zu geben, Texte barrierefrei zu verfassen und zu formatieren.

Danksagung

Wir möchten uns bei allen Personen bedanken, die uns bei unserer Arbeit unterstützt haben, um ein erfolgreiches Ergebnis zu liefern:

Inhaltsverzeichnis

1	Einleitung	4
1.1	Ausgangssituation	4
1.2	Ist-Zustand	4
1.2.1	Systemarchitektur	5
1.3	Motivation	6
1.3.1	Gleichstellung von Menschen mit Behinderung	6
1.4	Barrierefreiheit im Web	8
1.4.1	Definition: Web-Zugänglichkeit	8
1.5	Aufgabenstellung	9
1.6	Zielsetzung	9
1.7	Sollzustand	10
1.7.1	Erweiterte Systemarchitektur	10
1.7.2	Funktionale Anforderungen	11
1.7.3	Nichtfunktionale Anforderungen	11
1.8	Projektablauf und Produkt	11
1.8.1	Projektablauf	11
1.8.2	Produkt	12
1.9	Erfüllte Kriterien des WCAG 2.1	12
2	Ausgewählte Technologien	13
2.1	Git und GitLab	13
2.1.1	Versionskontrollsystem	13
2.1.2	Git	16
2.1.3	GitLab	19
2.2	Node.js und npm	20
2.3	Hypertext Markup Language	20
2.4	JavaScript	20
2.5	TypeScript	20
2.6	Webpack	20
3	Ausgewählte Aspekte	21
3.1	WCAG 2.1	21
3.2	Web Components	22

3.3	Factory Method Pattern	22
3.3.1	Was sind Design Patterns?	22
3.3.2	Begriffserklärung und Verwendung	23
3.3.3	Vorteile und Nachteile	24
3.4	JavaScript Events & EventListener	24
3.5	MutationObserver	26
3.6	Delegation	26
3.7	Barrierefreie Toolbar	26
3.7.1	Bedienbarkeit & Navigation	26
3.7.2	Toolbar Replacer Factory	26
3.7.3	Toolbar Replacer	26
3.8	Elemente der Toolbar	26
3.8.1	Gruppierung der Buttons	26
3.8.2	Button Elemente	26
4	Resümee	28
4.1	Halil Bahar	28
4.2	Sonja Cao	28
A	Arbeitsteilung	33
B	Protokolle	34

Kapitel 1

Einleitung

1.1 Ausgangssituation

Die Fa. Fabasoft ist ein mittelständisches Unternehmen im Zentralraum Oberösterreichs. Zum Geschäftsmodell zählen die Entwicklung und der Vertrieb von Softwareprodukten, die vor allem für die Bereiche Enterprise Search, Wissensmanagement, digitale Geschäftsprozesse und unternehmensübergreifende Zusammenarbeit in der Cloud konzipiert sind. Kunden sind private und öffentliche Auftraggeber, insbesondere des E-Governments, und Wirtschaftsunternehmen, die hohe Sicherheitsanforderungen stellen und hauptsächlich im deutschsprachigen Raum beheimatet sind. Ein Grundkonzept der Fabasoft Cloud ist Barrierefreiheit. Dieses Produkt wurde im Oktober 2019 als erste Web-Applikation von der Österreichischen Computer Gesellschaft mit dem WACA-Zertifikat (Web Accessibility Certificate Austria [1]) in der Stufe Silber ausgezeichnet.

1.2 Ist-Zustand

Ein Grundkonzept der Fabasoft Cloud ist Barrierefreiheit, Ziel ist es, dass die Benutzeroberfläche von allen Menschen, unabhängig ihrer Behinderungen, bedient werden kann. Oftmals wird in Softwareprodukten der Fa. Fabasoft die Eingabe von einfachen, formatierten Texten gefordert, die in einem WYSIWYG (What You See Is What You Get) Rich Text Editor erfasst werden. Übersetzt bedeutet "What You See Is What You Get" so viel wie "Was du siehst, ist das, was du bekommst". Der Text in einem WYSIWYG Editor wird bei der Ausgabe des Dokuments also genau so angezeigt, wie er bei seiner Bearbeitung aussieht.

Ein Beispiel für solch einen Editor ist "Trix". Er ist ein Open Source WYSIWYG Texteditor vom Unternehmen Basecamp, den Machern von Ruby on Rails, und wurde von den beiden Entwicklern Sam Stephenson und Javan Makhmali der Community auf GitHub [2] zugänglich gemacht. Mithilfe dieses Editors können Nachrichten, Kommentare, Artikel, Listen und vieles mehr verfasst werden. Es besitzt ein selbst entwickeltes Document Object Model (DOM), das neben den gängigen JavaScript Events ebenso auf selbst definierte Events hört und eingebettete Anhänge unterstützt.

1.2.1 Systemarchitektur

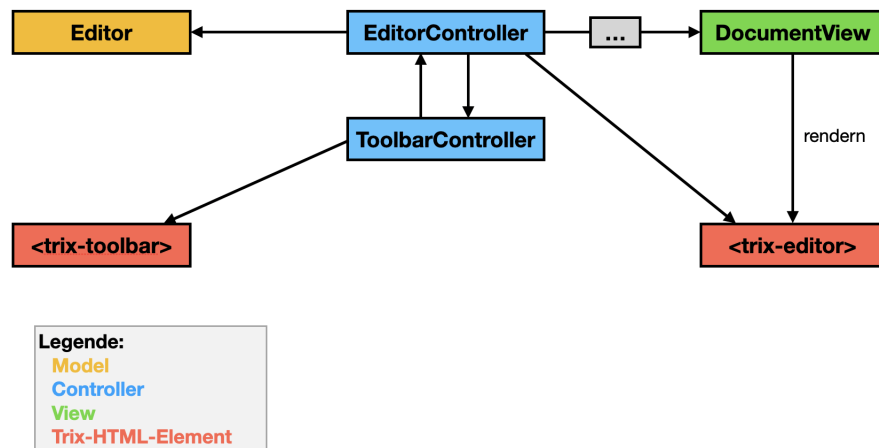


Abbildung 1.1: Vereinfachte Darstellung der Kommunikation zwischen den Klassen

Zur besseren Verständlichkeit wurden nur die Klassen dargestellt, die oberflächlich erkennbar sind. Trix unterscheidet sich bezüglich der Funktionalität von seiner Konkurrenz und ist deshalb um einiges komplexer. Außerdem verwendet er den Standard der Web Components (3.2). Sobald die Trix Library eingebunden ist, werden ihre benutzerdefinierten HTML-Elemente (`<trix-editor>`, `<trix-toolbar>`) registriert. Beim Verwenden eines Trix Editors werden dementsprechend der Controller und in diesem alle weiteren wichtigen Klassen instanziiert, die nicht unbedingt in der Darstellung ersichtlich sind.

EditorController

Dieser Controller ist eine der Klassen, die umfangreichere Funktionalitäten beinhalten. Er kümmert sich um

- **InputEvents** bei jeder Tastatureingabe im Editor. Hier wird beispielsweise darauf geachtet, dass beim Einfügen eines Textes dessen Formatierung weiterhin bestehen bleibt.
- Formatierungen bei Bedarf und das Aktivieren und Deaktivieren dieser, was aufgrund der Kommunikation zwischen den Controllern auch über die Toolbar möglich ist.
- das Rendering, wobei er es im `DocumentView` auslöst.

ToolbarController

Wie auch im Absatz davor erwähnt, kümmert sich dieser Controller um das Aktivieren und Deaktivieren von Formatierungen über die Buttons in der Toolbar. Er hat einen eigenen `EventListener` auf sich selbst (`<trix-toolbar>`). Die Besonderheit besteht darin, dass beliebig viele Buttons verwendet werden können. Der Grund dafür ist, dass das Event das sogenannte “Bubbling“ verwendet. Dies kann man sich wortwörtlich wie eine Blase unter Wasser vorstellen. Wenn ein HTML-Element ein Event besitzt, wird dieses Event auch von HTML-Elementen darunter (Child Elements) ausgelöst. Diese Blase steigt in der Struktur auf und löst das Event aus.

Die Buttons haben erst dann eine Funktion, wenn ein selbst definiertes HTML-Attribut hinzugefügt wird. Sobald das Event ausgelöst wird, überprüft der Controller, ob das Attribut vorhanden ist. Wenn das der Fall ist, handelt er anhand des Wertes und formatiert (kursiv) oder löst eine Funktion (rückgängig) aus.

DocumentView

Diese View ist für das gesamte Rendering im Editor zuständig. Das beinhaltet alle Textformatierungen und Textpositionierungen, die auch dementsprechend eine eigene View besitzen, wie etwa `TextView` für Texte oder `AttachmentView` für Bilder und Dokumente. Bei jedem Aufruf des Rederings wird das alte Dokument durch ein neues ersetzt.

1.3 Motivation

Der bisher integrierte Browser-UI-Control in Fabasoft Softwareprodukten weist einige Nachteile auf. Der sogenannte “CKEditor“ [4] in der Version 4 verliert seinen Long Term Support (LTS) im Jahr 2023. Dieser wird von der Version 5 [5] abgelöst. Da die neue Version eine neue Lizenz mit sich bringt, ist dieser Texteditor in den Produkten nicht mehr weiterhin verwendbar. Der Grund dafür ist, dass die neue Lizenz unter den Bedingungen der GNU General Public License Version 2 or later (GPLv2) steht und der Sourcecode auf Nachfrage freigegeben werden muss. Das entspricht nicht der Philosophie der Firma. Darum wird der Editor durch ein alternatives Produkt abgelöst. Durch eine erste Analyse hat sich herausgestellt, dass sich der Texteditor *Trix* als Ersatz eignet. Allerdings ist dieser nicht barrierefrei.

1.3.1 Gleichstellung von Menschen mit Behinderung

In der Gesellschaft und im Alltag machen Menschen mit Behinderung oftmals mit kleineren oder größeren Benachteiligungen Bekanntschaft. Global haben etwa 15 Prozent der Weltbevölkerung eine Art von Beeinträchtigung, wobei die rund eine Milliarden Kinder und Erwachsenen assistive Technologien, wie beispielsweise Rollstühle oder Hörgeräte, benötigen. International hat jedes Land eine eigene Definition für den Begriff, weshalb die WHO (World Health Organisation, vgl. [6]) diesen nur grob beschreiben kann. Sie geht dabei immer von drei Punkten aus:

- **Impairment (dt. Schädigung)** bezeichnet jede Anomalie oder jeden Verlust der anatomischen, psychischen oder physiologischen Funktionen und Strukturen des menschlichen Körpers.
- **Disability (dt. Beeinträchtigung)** bezeichnet jeden Mangel oder jede Einschränkung der Fähigkeiten infolge einer Beeinträchtigung, wodurch eine von der Gesellschaft als normal oder in diesem Bereich betrachtete Tätigkeit nicht ausgeübt werden kann.
- **Handicap (dt. Behinderung)** bezeichnet den Nachteil einer Person aufgrund einer Schädigung oder Beeinträchtigung. Eine Behinderung steht im Zusammenhang mit einem Problem mit einer Struktur oder eines Organes des Körpers.

Grundsätzlich wird eine Behinderung in Österreich als dauerhafte Auswirkung einer körperlichen, geistigen oder psychischen Beeinträchtigung oder Beeinträchtigung der Sinnesfunktionen bezeichnet. Diese überschreitet einen Zeitraum von sechs Monaten und erschwert die Teilhabe in der Gesellschaft. Damit die Benachteiligung von Behinderten möglichst flächendeckend beseitigt wird, können diesbezügliche Gesetze auf der Webseite des Rechtsinformationssystems des Bundes (RIS) abgerufen werden:

- Das Bundesgesetz über die Gleichstellung von Menschen mit Behinderungen (Bundesbehindertengleichstellungsgesetz - BGStG, vgl. [7]) zur Regelung der Diskriminierung im täglichen Leben.
- Das Behinderteneinstellungsgesetz (BEinstG, vgl. [8]) zur Regelung der Diskriminierung in der Arbeitswelt.
- Das Bundesgesetz vom 17. Mai 1990 über die Beratung, Betreuung und besondere Hilfe für behinderte Menschen (Bundesbehindertengesetz - BBG, vgl. [9]) zur Regelung der Aufgaben und Befugnisse des Behindertenanwalts.

Die Fassung des BGStG besagt, dass das Ziel die Beseitigung der Diskriminierung von Menschen mit Behinderungen sei, um diese gleichberechtigt am Leben in der Gesellschaft teilhaben zu lassen und eine selbstbestimmte Lebensführung zu gewährleisten. Der Diskriminierungsschutz gilt somit für diejenigen, die körperlich, geistig, psychisch behindert oder sinnesbehindert sind, ebenso für ihre Angehörigen. Es ist nicht erlaubt jemanden unmittelbar oder mittelbar zu diskriminieren.

Definition: Diskriminierung

Wird eine Person aufgrund ihrer Behinderung in bestimmten Situationen oder gegenüber anderen anders behandelt, so liegt eine unmittelbare Diskriminierung vor. Eine mittelbare Diskriminierung hingegen liegt genau dann vor, wenn neutrale Vorschriften, Kriterien, Verfahren oder Merkmale gestalteter Lebensbereiche den Anschein erwecken, Menschen mit Behinderungen gegenüber anderen auf jegliche Art und Weise zu benachteiligen. Zu diesen Benachteiligungen gehören:

- Barrieren
- eine weniger günstige Behandlung
- Diskriminierungsaufforderungen
- und Belästigung aufgrund einer Behinderung

Arten von Behinderungen

Grundsätzlich lassen sich Behinderungen in sechs Kategorien einteilen:

- **Körperliche Behinderungen** kommen am häufigsten vor und nehmen nicht selten mit dem Alter zu. Bezeichnet werden damit starke physische Einschränkungen, die auf eine Dysfunktion oder Schädigung der Stütz- und Bewegungsorgane zurückgeführt werden.
- **Geistige Behinderungen** sind fortwährende, deutlich über dem Durchschnitt liegende Einschränkungen der kognitiven Fähigkeiten und kommen am zweithäufigsten vor. Zu diesen kognitiven Fähigkeiten zählen die Wahrnehmung, die Aufmerksamkeit, das Denken und das Lernen ebenso wie die Erinnerung, die Motivation und die Konzentration.
- **Sinnesbehinderungen** umfassen alle Seh- sowie Hörbehinderungen wie Fehlsichtigkeit, Blindheit, Schwerhörigkeit, Gehörlosigkeit und Taubblindheit.
- **Sprachbehinderungen** oder anders formuliert Störungen des Redeflusses, des Spracherwerbs, des Sprechens und der Stimme bezeichnet all jene Menschen, die nicht fähig sind ihre Muttersprache im entsprechenden Alter schriftlich oder mündlich zu beherrschen.
- **Psychische Behinderungen** umfassen die Beeinflussung des Denkens, Fühlens und Handelns von Menschen. Das heißt wiederum, dass es Abgrenzungen zwischen Verhalten und Erleben gibt.

1.4 Barrierefreiheit im Web

1.4.1 Definition: Web-Zugänglichkeit

Web-Zugänglichkeit oder im Englischen auch “Web Accessibility” bezeichnet, dass alle Menschen auf dem bestmöglichen Weg Zugang zu Information oder Dienstleistung im Internet haben. Ziel ist es, Websites und jegliche mobile Anwendungen für alle Nutzerinnen und Nutzer, vor allem auch diejenigen mit Behinderungen, besser zugänglich und somit barrierefrei zu gestalten. In der heutigen Gesellschaft ist das Internet mit all seinen verfügbaren Leistungen kaum noch wegzudenken. Zahlreiche Dienstleistungen wie etwa Onlinebanking, elektronische Dienste von Instituten und Institutionen, Onlineshop und viele mehr gehören dazu.

Das Web-Zugänglichkeits-Gesetz (WZG, vlg. [10]) umfasst alle rechtlichen Grundlagen zur Erreichung dieses Ziels. Insbesondere der Bund und alle mit ihm in Zusammenhang stehende Einrichtungen sind dazu verpflichtet barrierefreie Dienstleistungen bereitzustellen. Hierzu gibt es bestimmte Anforderungen:

- Wahrnehmbarkeit
- Bedienbarkeit
- Verständlichkeit
- Robustheit

Des Weiteren wurde ein zeitlicher Anwendungsbereich festgelegt, in dem alle Websites und mobile Anwendungen des Bundes barrierefrei sein müssen:

- Neue Websites nach dem 23. September 2018 ab dem 23. September 2019
- Alte Websites vor dem 23. September 2018 ab dem 23. September 2020
- Alle mobilen Anwendungen ab dem 23. Juni 2021

Damit diese Dienstleistungen fortlaufend zugänglich sind, müssen die oben genannten Anforderungen regelmäßig überprüft werden. Grundsätzlich hat jede Nutzerin und jeder Nutzer das Recht dazu, sich über Mängel bei Nichteinhaltung der Barrierefreiheit zu beschweren, die dann, wenn sie berechtigt sind, beseitigt werden müssen.

1.5 Aufgabenstellung

Im Rahmen dieser Diplomarbeit wurde eine Erweiterung für den Texteditor *Trix* geschrieben, sodass dieser den Ansprüchen für die Produkte der Fa. Fabasoft gerecht wird. Insbesondere wurden die Kriterien des WCAG 2.1 (Web Content Accessibility Guidelines 3.1) erfüllt sowie eine Bedienung mit Screenreadern und vollständig mit Tastatur ermöglicht worden.

1.6 Zielsetzung

Ziel dieser Arbeit war es, den Mitarbeitern und Kunden von der Fa. Fabasoft die Verwendung des Systems weiterhin zu ermöglichen. Der Texteditor sollte weitestgehend von allen Personen einfach und ohne Probleme bedient werden können.

1.7 Sollzustand

1.7.1 Erweiterte Systemarchitektur

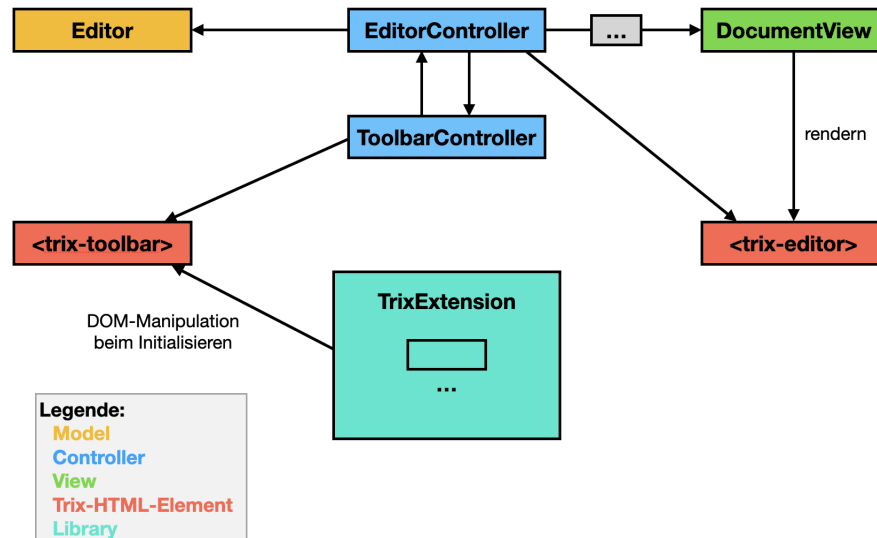


Abbildung 1.2: Vereinfachte Darstellung der Kommunikation zwischen den Klassen mit der Erweiterung

Mit der Erweiterung “Trix Extension“ ist es möglich, die Toolbar in wenigen Codezeilen robust und sicher zu ersetzen. Hierbei wird der gesamte Inhalt von dem `<trix-toolbar>` überschrieben. Die Toolbar bestand zuvor aus HTML-Elementen, die statisch in einer JavaScript-Datei definiert waren. Die Struktur der neu ersetzten Toolbar bleibt dennoch dieselbe für Backwards Compatibility. Zusätzlich zu der neuen Anordnungs- und Designmöglichkeit der Buttons werden ARIA-Rollen (Accessible Rich Internet Applications / WAI-ARIA, vgl. [?]) zugeordnet. Des Weiteren werden `KeyEvents` an `<trix-editor>` und `<trix-toolbar>` gehängt. Das Event für den Editor ist sehr simpel, denn er hört auf nur eine Tastenkombination, die es ermöglicht den Fokus auf die Toolbar zu legen. Bei der Toolbar reagiert das Event nur bei Tasten, wie

- Pfeiltasten zum Navigieren zwischen den Buttons,
- Pfeiltasten mit Steuerung-Taste zum Navigieren zwischen den Button-Gruppen,
- Escape-Taste zum Verlassen der Toolbar und zurück Fokussieren des Editors,
- Pos1- und Ende-Taste zum Navigieren zum jeweiligen ersten und letzten Button,

- und Tab-Taste ignorieren, damit die Nutzer nicht verwirrt werden und die Orientierung verlieren.

1.7.2 Funktionale Anforderungen

Für diese Diplomarbeit sind keine funktionalen Anforderungen definiert. Der Grund ist der, dass die Autoren der Meinung sind, dass es keine neuen funktionalen Anforderungen gibt, weshalb die Arbeit schwerwiegend auf nichtfunktionalen Anforderungen basiert.

1.7.3 Nichtfunktionale Anforderungen

- Aufsetzen eines npm-basierten Build-Prozesses mit allen notwendigen Artefakten in das Fabasoft npm-Repository.
- Aufbau einer Unit-Test-Infrastruktur mit einer Line-Coverage von mindestens 70 Prozent.
- Einpflegen der entsprechenden Änderungen zur Erfüllung der Kriterien des WCAG 2.1.
- Vollständige Bedienung über eine Tastatur und einen Screenreader.
- Einpflegen eines High-Level-APIs, um die Instanziierung und Parametrierung aus dem Fabasoft UI möglichst simpel zu gestalten.

1.8 Projektablauf und Produkt

1.8.1 Projektablauf

Im ersten Schritt ist das GitHub Repository des Texteditors Trix geforkt worden, um die notwendigen Änderungen zur Erreichung der Barrierefreiheit einzupflegen. Allerdings sind die Entwickler, die auch die gleichzeitigen Betreuer des Editors sind, bezüglich der Issues und der Pull Requests, sehr inaktiv. Daraufhin wurde mit dem Team der Fabasoft Cloud entschieden, dass eine Erweiterung für den Texteditor sinnvoller ist.

Im nächsten Schritt wurde der korrekte Export der bestehenden formatierten Textinhalte in der Fabasoft Cloud ermöglicht. Damit keine zusätzlichen Zeilenumbrüche hinzugefügt und gewisse HTML Elemente unterstützt und nicht übersprungen werden, sind eine Reihe von Funktionalitäten in Trix selbst und mit Hilfe von JavaScript Prototypes überschrieben worden. Bei Bedarf können diese Überschreibungen ignoriert werden.

Während der Entwicklungsphase dieser Diplomarbeit revidierte ein blinder Mitarbeiter regelmäßig die Funktionen des Texteditors. Zusätzlich zu den Verbesserungsvorschlägen teilte er in seinen Feedbacks mit, welche Kriterien des WCAG 2.1 gut umgesetzt wurden und welche noch erforderlich waren.

1.8.2 Produkt

Mithilfe der Erweiterung ist es möglich, den Texteditor über die Tastatur zu bedienen und somit visuelle Elemente, wie etwa Buttons, zum Formattieren des Textes zu erreichen.

1.9 Erfüllte Kriterien des WCAG 2.1

Kapitel 2

Ausgewählte Technologien

2.1 Git und GitLab

2.1.1 Versionskontrollsystem

Oftmals arbeitet ein Entwickler oder ein Entwicklerteam an einem Softwareprojekt, um entweder Fehler zu beheben oder neue Funktionalitäten hinzuzufügen. Hierbei werden Änderungen am Quellcode durchgeführt, die regelmäßig gesichert werden müssen. Diese Änderungen können in manchen Fällen auch dazu führen, dass etwas anderes im Code nicht mehr so funktioniert wie es sollte. Dann werden weitere Änderungen gemacht, um wieder einen lauffähigen Stand herzustellen, jedoch kann es passieren, dass plötzlich gar nichts mehr funktioniert.

Abhilfe für dieses Problem bietet ein Versionskontrollsystem, auch als VCS (Version Control System, vgl. [11]) bezeichnet, das Änderungen in der Projektentwicklung festhält. Dadurch ist es möglich zu einem späteren beliebigen Zeitpunkt auf ältere Versionen des Systems zurückzugreifen, den aktuellen Code mit früheren Versionen zu vergleichen oder Bugfixes zu implementieren. Arbeiten mehrere Entwickler an denselben Dateien im Quellcode, so kann mitverfolgt werden, welche Person welche Änderungen gemacht hat. Ein Projekt, das mit einem VCS verwaltet wird, heißt Repository. Ein Repository kann als eine Art Datenbank mit allen Änderungen betrachtet werden, die die History repräsentieren. Diese Änderungen sind ein markierter Stand, ein sogenannter Commit, in der Projektentwicklung und werden als Working Copy gespeichert, die einer Kopie des gesamten Projektes entspricht. Ein Commit kann als Schnappschuss oder als kleiner Meilenstein betrachtet werden. Er beinhaltet neben der Working Copy einen Zeitstempel und eine Nachricht, die aussagt, was sich seit dem letzten Commit geändert hat. Die Entwicklungsarbeiten können dabei in mehreren Entwicklungszweigen (engl. Branches) erfolgen. Eine Abspaltung eines Projektes wird hingegen als Fork bezeichnet.

Es gibt drei Arten von Version Control Systems:

- **Lokal:** Lokale Versionskontrollsysteme funktionieren nur auf einem Rechner und

versionieren eine einzige Datei. Insbesondere in Büroanwendungen kommen Tools wie Revision Control System (RCS) oder Source Code Control System (SCCS) zum Einsatz. Das Dokument speichert dabei jede Version in seiner eigenen Datei.

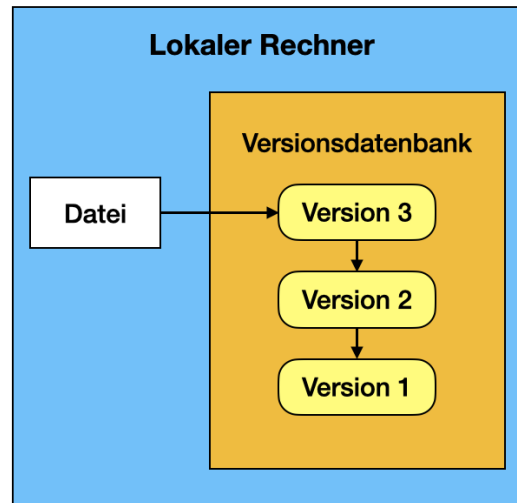


Abbildung 2.1: Lokales Versionskontrollsystem

- **Zentral:** Bei einem zentralisierten Versionskontrollsystem gibt es ein Repository, das sich mehrere Entwickler in einem Netzwerk miteinander teilen. Es handelt sich hierbei um ein Client-Server-System, das von zahlreichen kommerziellen Anbietern verwendet wird. Durch das Concurrent Versions System (CVS) wurde dieses Konzept berühmt, aber durch Subversion (SVN) neu implementiert.

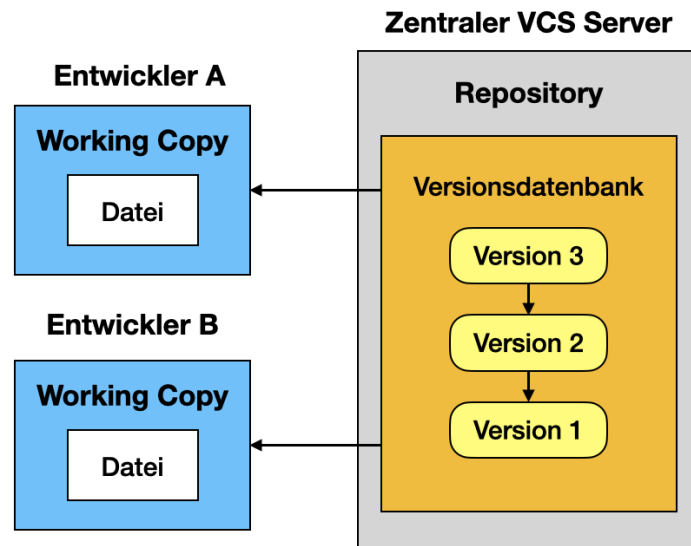


Abbildung 2.2: Zentralisiertes Versionskontrollsystem

- **Verteiltes VCS:** Im Gegensatz zum zentralisierten VCS existiert beim verteilten Versionskontrollsystem kein zentrales Repository. Jeder am Projekt tätige Entwickler verfügt über ein eigenes Repository, das er mit anderen Repositories abgleichen kann. Auch hier ist die History klar ersichtlich. Der einzige Unterschied zu den beiden anderen Versionskontrollsystemen ist der, dass Änderungen hier am lokalen Rechner erfolgen können. Eine Verbindung mit dem Server ist nicht notwendig.

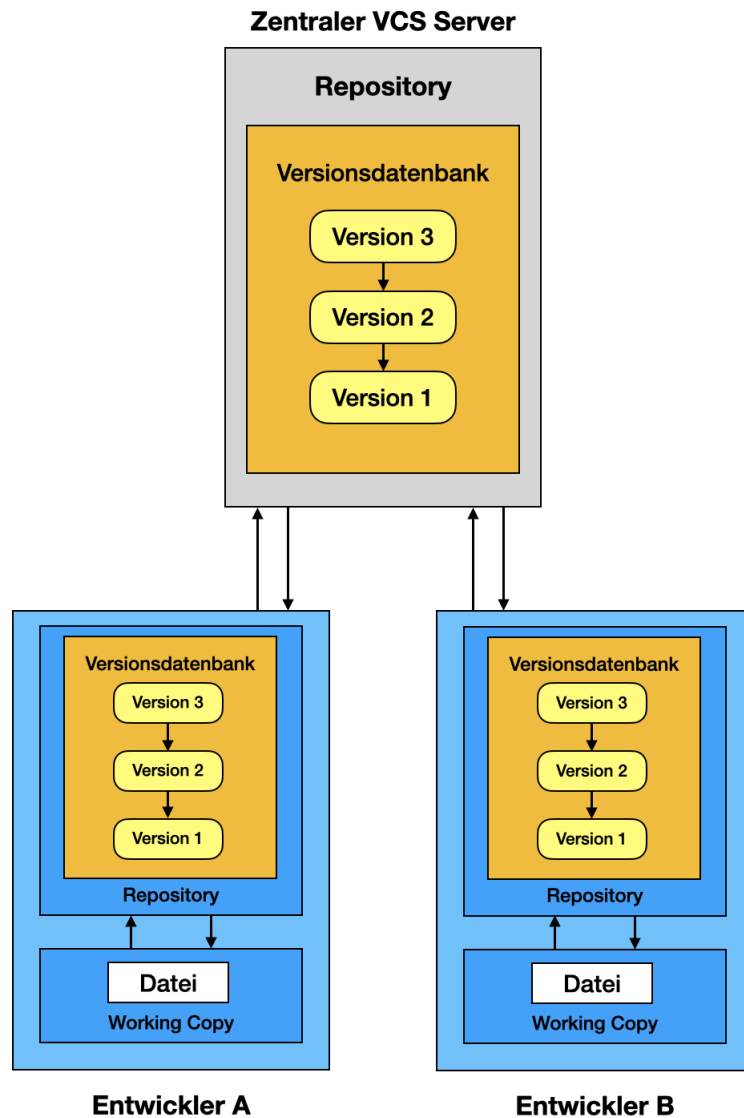


Abbildung 2.3: Verteiltes Versionskontrollsystem

2.1.2 Git

Git [12] ist ein Open-Source-Tool von Linus Torvalds, dem Entwickler des Betriebssystems Linux, aus dem Jahr 2005. Heutzutage zählt es zu eines der weitverbreitesten Versionsverwaltungssysteme weltweit. Die Softwareentwickler kommen sowohl aus dem kommerziellen als auch aus dem öffentlichen Bereich. Aufgrund seiner Architektur ist Git ein Distributed VCS (DVCS, dt. verteiltes VCS) und funktioniert in zahlreichen Plattformen und Entwicklungsumgebungen. Das bedeutet, dass auf die gesamte History

mit allen Entwicklungsarbeiten von jedem Standort aus zugegriffen werden kann. Neben seinem verteilten System ist Git unter anderem auf Performance, Sicherheit und Flexibilität fokussiert.

Snapshots

Bei den meisten VCS werden Informationen als eine Liste mit den Änderungen innerhalb der Dateien gespeichert. Das bedeutet, dass bei jeder Version nur die Dateien festgehalten werden, deren Inhalt sich während den Entwicklungsarbeiten geändert haben.

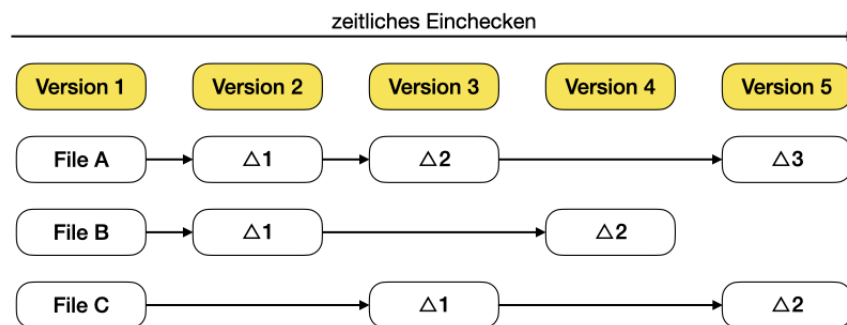


Abbildung 2.4: Speichern von Daten bei anderen VCS

Im Gegensatz dazu werden die Daten als eine Art Folge von Schnappschüssen (engl. snapshots) betrachtet. Nach jedem Commit wird der aktuelle Stand des Repositories gespeichert. Dieser beinhaltet die gesamte Kopie des Projekts zu diesem Zeitpunkt und erhält eine Referenz zu diesem Schnappschuss in Form eines einzigartigen Hashwertes. Wenn das Entwicklerteam weiter am Projekt arbeitet und die Änderungen committet, erhalten die neuen und geänderten Dateien einen neuen Hashwert, während die Dateien, die sich nicht geändert haben, nur auf dieselbe Datei im vorherigen Schnappschuss verlinkt werden.

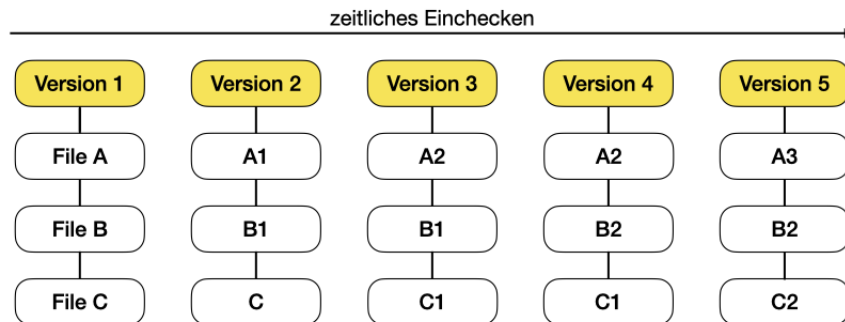


Abbildung 2.5: Speichern von Daten bei Git

Um eine Sicherheit garantieren zu können, werden die Dateiinhalte sowie ihre Beziehungen zu anderen Dateien, Verzeichnissen, Versionen, Commits und Tags mit dem Hashing-Algorithmus SHA1 gesichert. Dadurch kann der Quellcode des Entwicklers vor ungewollten Änderungen geschützt und die Historie vollständig mitverfolgt werden. In der Datenbank werden keine Dateinamen sondern nur Hashwerte gespeichert. Ein Beispiel für so einen Wert ist `fe215ed7198155ca796fbb8ef81683137c210492`.

Workflow

In einem Git-Projekt sind die drei wichtigsten Stufen das Working Directory, die Staging Area und das Repository, wobei dieses in das `local` und das `remote` Repository gegliedert werden kann. Mit `add` wird eine Kopie des Working Directory erstellt, wobei die Änderungen an den Dateien mit `new`, `modified` oder `deleted` gekennzeichnet sind. Die Arbeitskopie befindet sich nun in der Staging Area und noch nicht in der Datenbank. Nun haben die Dateien den Status `staged` und können im nächsten Schnappschuss eingepflegt werden. Nach einem `commit` wird ein Schnappschuss im lokalen Repository erstellt, der mit einem `push` in der Datenbank und somit im entfernten Repository gesichert wird.

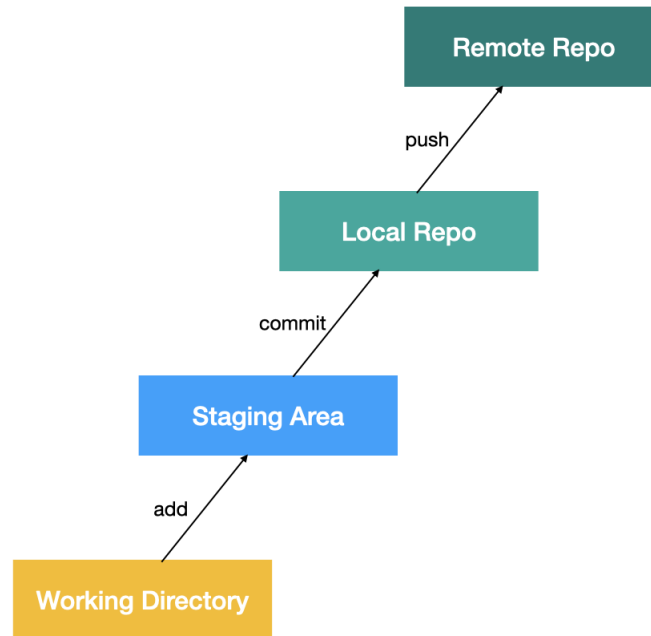


Abbildung 2.6: Die drei Hauptstufen von Git

2.1.3 GitLab

Im Jahr 2011 wurde GitLab von den ukrainischen Entwicklern Dmitri Saparoschcz und Valery Sizov ursprünglich in der Programmiersprache Ruby on Rails und mittlerweile noch zusätzlich in Go und Vue.js geschrieben. Es ist eine Webapplikation zur Versionskontrolle, das auf Git basiert. Neben dem Arbeiten mit Git stellt diese Anwendung einige weitere Funktionen zur Verfügung. Diese sind beispielsweise eine integrierte und kostenlose Continuous Integration/Delivery (CI/CD), das Issue Tracking, die Wiki-Funktion und viele mehr. Zusätzlich können Rollen zugewiesen werden, die die Berechtigungen von den jeweiligen Nutzern festlegen.

Seit 2013 gibt es zwei Lizenzmodelle von GitLab. Die Community Edition (CE) wird unter der MIT-Lizenz als Open-Source-Software entwickelt und ist auf gitlab.com als Software as a Service (SaaS) erreichbar. Hingegen wird die Enterprise Edition unter einer proprietären Lizenz entwickelt und beinhaltet mehr Funktionen, die für Unternehmen relevanter sind. Bei der EE können Repositories auf einem eigenen Server verwaltet werden.

- 2.2 Node.js und npm**
- 2.3 Hypertext Markup Language**
- 2.4 JavaScript**
- 2.5 TypeScript**
- 2.6 Webpack**

Kapitel 3

Ausgewählte Aspekte

3.1 WCAG 2.1

Die Web Content Accessibility Guidelines 2.1 (WCAG 2.1, vgl. [13]) wurden im Dezember 2008 veröffentlicht und sind ein Standard, der weltweit verwendet wird, um Dienstleistungen im Web so barrierefrei wie möglich zu gestalten. Somit können auch Menschen mit Behinderungen, wie etwa Sehschwäche, Blindheit, Hörverlust, Taubheit, körperliche oder kognitive Einschränkung, Sprachbehinderung, Lichtempfindlichkeit, Lernbehinderung oder Kombinationen dieser, das World Wide Web nutzen. Diese Richtlinien sind vom World Wide Web Consortium (W3C, vgl. [14]) am 5. Juni 2018 für Webanwendungen empfohlen worden.

Durch die Befolgung der Empfehlungen des W3C können Webdesigner und -entwickler, politische Entscheidungsträger, Käufer, Lehrer und Studenten beziehungsweise Schüler das WWW nahezu problemlos nutzen. Die WCAG 2.1 setzen sich aus allgemeinen Prinzipien, allgemeinen Richtlinien, prüfbaren Erfolgskriterien sowie einer Vielzahl an Techniken zusammen:

- Es gibt vier **Prinzipien**, die die Grundlagen für die Web-Zugänglichkeit bilden: wahrnehmbar, bedienbar, verständlich und robust (englisch: perceivable, operable, understandable and robust).
- Die Prinzipien beinhalten insgesamt 13 **Richtlinien**, an die sich Entwickler halten sollen, um das Web zugänglicher zu machen. Sie sind zwar nicht prüfbar, helfen allerdings dabei die Erfolgskriterien besser zu verstehen und die Techniken besser umzusetzen.
- **Prüfbare Erfolgskriterien** werden dort eingesetzt, wo bestimmte Anforderungen und Tests für die Konformität erforderlich sind, wie etwa beim Entwurf, beim Kauf, bei Regelungen und bei vertraglichen Vereinbarungen. Es gibt drei Ebenen der Konformität, wobei die höchste AAA, die mittlere AA und die niedrigste A ist.

- **Ausreichende und beratende Techniken** dienen zur Erfüllung der Erfolgskriterien und als Beratung. Häufig vorkommende Misserfolge sind ebenfalls dokumentiert.

Bei Erreichung der höchsten Ebene ist eine hundertprozentige Web-Zugänglichkeit jedoch nicht garantiert. Die WCAG 2.1 dienen ausschließlich als Anleitung, um das Web so barrierefrei wie möglich zu gestalten. Deshalb wird vom W3C empfohlen sich regelmäßig über den neuesten Stand zu informieren und sich mit anderen zu beraten. Folgende Bemerkung wurde vom W3C (2008, [13]) auf der Webseite veröffentlicht: “[...] Authors are encouraged to consider the full range of techniques [...] as well as to seek relevant advice about current best practice to ensure that Web content is accessible, as far as possible, to this community. [...]“

3.2 Web Components

Web Components (vgl. [3]) sind eine Web-Technologie, die es den Entwicklern ermöglicht, selbstdefinierte HTML-Elemente zu erstellen und wiederzuverwenden. Diese sind mit ihrem CSS und JavaScript gekapselt und sind somit vollständig von anderem Code getrennt. Mit einigen JavaScript Frameworks wie etwa Angular war es zwar bereits möglich wiederverwendbare HTML-Elemente zu definieren, allerdings nutzt jedes der Frameworks einen anderen Standard. Dies bedeutet, dass der Code in anderen Projekten in den meisten Fällen nicht verwendbar ist. Genau dieses Problem wurde durch Web Components gelöst. Die Web Components bestehen aus drei Hauptbestandteilen:

- **Customs ELeMENTS:** Ein Satz von JavaScript APIs zur Definition von benutzerdefinierten Elementen.
- **Shadow DOM:** Ein Satz von JavaScript APIs zum Hinzufügen eines DOM-Elementes mit gekapselten Shadow-DOM-Elementen, welches separat vom Hauptdokument DOM gerendert wird. Dadurch ist es möglich jegliche Funktionalitäten isoliert zu definieren, sodass der Programmierer keine Rücksicht auf Kollisionen mit anderen Dokumenten nehmen muss.
- **HTML Templates:** Markup-Vorlagen, die innerhalb der Elemente `<template>` und `<slot>` geschrieben werden, werden nicht auf der dargestellten Seite abgebildet. Diese sind dafür da, um sie mehrmals als Vorlage eines benutzerdefinierten Elements zu verwenden.

3.3 Factory Method Pattern

3.3.1 Was sind Design Patterns?

Um das Factory Method Pattern zu verstehen, muss man zunächst einmal wissen, was ein Design Pattern (dt. Entwurfsmuster, vgl. [15]) ist. In der Softwareentwicklung treten

beim Entwurf oftmals dieselben Probleme auf. Abhilfe hierfür schaffen Design Patterns, die eine allgemeine wiederholbare Lösung dieser Designprobleme sind, den Entwicklungsprozess beschleunigen und programmiersprachenunabhängig sind. Sie repräsentieren somit eine Idee, und keine spezielle Implementierung, durch deren Verwendung der Quellcode flexibler, wiederverwendbar und einfacher für die Entwickler zu warten ist. Allerdings sind sie nicht in jedem Projekt zwingend erforderlich, da sie lediglich für die Problemlösung und nicht für die Projektentwicklung gedacht sind.

Arten von Design Patterns

- **Creational Pattern (dt. Erzeugungsmuster):** Sie dienen dazu Objekte zu erzeugen. Dieser Prozess wird gekapselt und ausgelagert, sodass die Objekterzeugung von der Implementierung des Objektes klar getrennt wird.
- **Structural Pattern (dt. Strukturmuster):** Sie stellen vorgefertigte Vorlagen für die Beziehungen zwischen den Klassen zur Verfügung und erleichtern somit den Softwareentwurf.
- **Behavioral Pattern (dt. Verhaltensmuster):** Sie führen dazu, dass die Software flexibler wird, indem das komplexe Verhalten dieser Software modelliert wird.

3.3.2 Begriffserklärung und Verwendung

Das Factory Method Pattern (vgl. [16]), auch bekannt als Factory Pattern oder Fabrikmethode, definiert ein Interface oder eine abstrakte Klasse zur Erstellung von Objekten. Die Objekterstellung wird hierbei von den Subklassen übernommen, die entscheiden welche Klasse instanziiert wird.

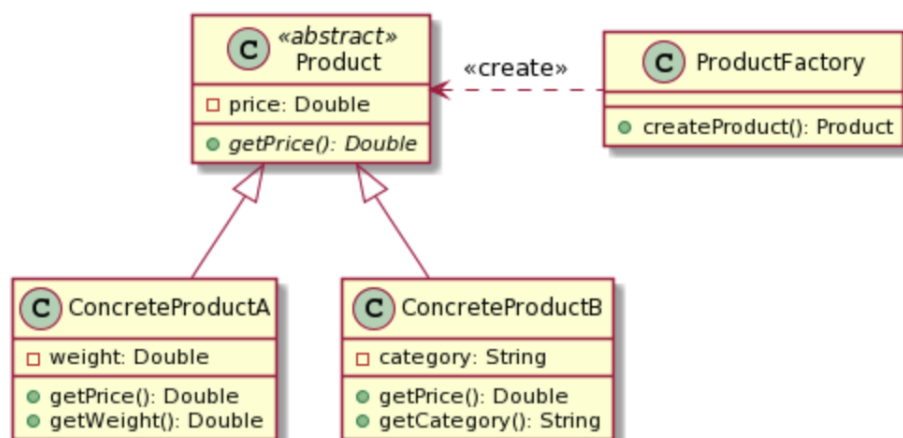


Abbildung 3.1: Veranschaulichung des Factory Method Patterns in einem Klassendiagramm

3.3.3 Vorteile und Nachteile

Vorteile

- Mit Hilfe des Factory Method Patterns können Unterklassen die Art, wie Objekte erstellt werden sollen, selbst wählen.
- Dieses Design Pattern fördert die lose Kopplung. Das heißt im Allgemeinen, dass anwendungsspezifische Klassen nicht mehr in den Code eingebunden werden müssen.
- Die Kommunikation erfolgt ausschließlich über die Schnittstelle oder die abstrakte Klasse, so dass der Code mit allen Klassen funktioniert, die entweder die Schnittstelle implementieren oder die abstrakte Klasse erweitern.

Nachteile

- Falls die verwendeten Klassen nicht abstrakt sind, muss für jede individuelle Klasse eine Factory geschrieben werden.

3.4 JavaScript Events & EventListener

trix-initialize

Sobald der `<trix-editor>` im DOM registriert wird und sein internes `Editor`-Objekt zur Benutzung bereit ist, wird das Event `trix-initialize` gefeuert.

Damit nicht die standardmäßig definierte Toolbar, sondern die erweiterte barrierefreie Toolbar, die diese ersetzen soll, angezeigt wird, ist es möglich auf dieses Event zu hören und dementsprechend darauf zu reagieren. Somit kann mit der Trix-Extension die Toolbar ersetzt werden, ohne dass der Endbenutzer etwas von diesem DOM-Manipulationsprozess mitbekommt.

mousedown

Das `mousedown` Event wird genau zu dem Zeitpunkt gefeuert, an dem sich der Mauszeiger auf einem HTML-Element befindet und dieses drückt.

Damit ein HTML-Button in der `<trix-toolbar>` aktiv ist und geklickt werden kann, hört dieser standardmäßig auf ein `mousedown` Event. Um diese Buttons mit einer Tastatur bedienen zu können, müsste der Trix-Editor im Quellcode auf Tastatureingaben reagieren und diese dementsprechend erneut validieren. In der Erweiterung Trix-Extension wird dieses Event künstlich ausgelöst mit `EventTarget.dispatchEvent()` und somit sind die Buttons in der Toolbar auch über die Tastatur bedienbar, ohne dass der Quellcode mit Wiederholung der Validation ergänzt werden muss.

keydown

Wenn eine beliebige Taste gedrückt ist, wird das `keydown` Event gefeuert und liefert einen Code, der aussagt, welche Taste im Moment gedrückt wird.

Um den WYSIWYG Texteditor mittels einer Tastatur bedienen zu können, wird auf das

keydown Event gelauscht. Je nachdem, welche Taste gedrückt wird, wird entweder der nächste oder der vorherige Button oder die Button-Gruppe fokussiert, geklickt oder der Fokus zur weiteren Texteingabe in den Editor gesetzt.

Folgende Tasten und Tastenkombinationen gelten in der Toolbar und im Editor:

Tasten	Fokusbereich	Beschreibung
ALT + F10	<trix-editor>	Befindet sich der Fokus im Editor, dann gelangt man mit dieser Tastenkombination in die Toolbar und der erste vorkommende Button wird fokussiert.
→ oder ↓	<trix-toolbar>	Der nächste bzw. rechte Button in der Toolbar wird fokussiert.
← oder ↑	<trix-toolbar>	Der vorherige bzw. linke Button in der Toolbar wird fokussiert.
STRG + →/↓	<trix-toolbar>	Der erste Button der nächsten bzw. rechten Gruppe wird fokussiert.
STRG + ←/↑	<trix-toolbar>	Der erste Button der vorherigen bzw. linken Gruppe wird fokussiert.
ENTER oder LEERTASTE	<trix-toolbar>	Der aktuell fokussierte Button wird geklickt.
ESC	<trix-toolbar>	Befindet sich der Fokus in der Toolbar und wird die Taste gedrückt, so wird wieder der Editor fokussiert und der Cursor befindet sich an der zuletzt verwendeten Position.
POS1	<trix-toolbar>	Der erste in der Toolbar vorkommende Button wird fokussiert.
ENDE	<trix-toolbar>	Der letzte in der Toolbar vorkommende Button wird fokussiert.

Tabelle 3.1: Tastenkombinationen zur Verwendung der Toolbar mit einer Tastatur

focusout

Sobald ein HTML-Element im Begriff dabei ist den Fokus zu verlieren, wird das **focusout** Event gefeuert.

Beim Initialisieren wird für jeden HTML-Button, der ein **DropDownButton** 3.8.2 ist, ein Dropdown Menü erstellt. Dieses Menü besteht aus einem HTML **<div>** Element, in dem sich weitere HTML-Buttons befinden. Damit es für den Benutzer zu Beginn noch nicht sichtbar ist, erhält es das Attribut **hidden**. Sobald der **DropDownButton** geklickt wird, wird auch das Menü sichtbar und der Fokus auf den ersten Button darin gelegt. Wenn das Menü allerdings nicht mehr fokussiert ist, also das Event **focusout** gefeuert wird, erhält das Dropdown Menü erneut das Attribut **hidden** und wird somit für den Benutzer

nicht mehr sichtbar sein.

3.5 MutationObserver

Der `MutationObserver` ist ein Interface, der Veränderungen in der Baumstruktur des DOMs beobachtet und wurde konzipiert, um die Mutation Events aus der DOM3 Events Spezifikation abzulösen.

Sobald die Toolbar geladen wird bzw. bestimmte Buttons geklickt werden, werden einige andere Buttons deaktiviert und erhalten das Attribut `disabled`. Dieses verhindert allerdings das Fokussieren eines Buttons, was mit dem Attribut `aria-disabled` problemlos funktioniert. Abhilfe verschafft deshalb ein `MutationObserver`. Erhält ein Button nun das Attribut `disabled`, beobachtet der `MutationObserver` diese Veränderung und entfernt es. Stattdessen fügt es das Attribut `aria-disabled` hinzu und setzt es auf `true`. Bei allen anderen Buttons, die nicht deaktiviert sind, ist `aria-disabled=false`.

3.6 Delegation

Die Delegation findet in der objektorientierten Programmierung verschieden Verwendung zur dynamischen Bindung von Methoden zur Programmlaufzeit.

Jedem `AttributeButton` kann die Identifikation eines HTML-Elements mitgegeben werden, das einen Dialog repräsentiert und bereits im HTML-Code erstellt wurde. Die Delegation ermöglicht eine Kommunikation zwischen der Trix-Toolbar und dem Dialog. Sie dient ausschließlich dazu den Dialog zu öffnen und zu schließen, wobei zwei zusätzliche Funktionalitäten das Erstellen und das Entfernen von Links sind. Alles, was während oder nach dem Dialog geschieht, wird vom Entwickler selbst bestimmt.

3.7 Barrierefreie Toolbar

3.7.1 Bedienbarkeit & Navigation

3.7.2 Toolbar Replacer Factory

3.7.3 Toolbar Replacer

3.8 Elemente der Toolbar

3.8.1 Gruppierung der Buttons

3.8.2 Button Elemente

Es werden vier verschiedene Arten von Buttons unterschieden:

- **Action Button**
- **Attribute Button**

- Clickable Button
- Dropdown Button

Action Button

Attribute Button

Clickable Button

Dropdown Button

Kapitel 4

Resümee

4.1 Halil Bahar

4.2 Sonja Cao

Literaturverzeichnis

- [1] Zertifikate WACA. Web Accessibility Certificate Austria (WACA) - Österreichs erstes Qualitätssiegel um Barrierefreiheit im Web nach den internationalen W3C-Richtlinien nach außen erkennbar zu machen. URL: <https://waca.at/zertifikate>.
- [2] Trix. Trix is a rich text editor for everyday writing developed by Javan Makhmali and Sam Stephenson. URL: <https://github.com/basecamp/trix>.
- [3] Web Components. Webkomponenten sind eine Gruppe von Web-Technologien, die es ermöglichen, benutzerdefinierte, wiederverwendbare HTML Elemente zu erstellen, deren Funktionalität gekapselt ist und damit vollständig getrennt von anderem Code. URL: https://developer.mozilla.org/de/docs/Web/Web_Components.
- [4] Smart WYSIWYG HTML editor CKEditor 4. Fully customizable WYSIWYG HTML editor with rich text features. Enterprise-grade with 70 languages and the approval of millions. URL: <https://ckeditor.com/ckeditor-4/>.
- [5] Rich text editor CKEditor 5. Easy to customize rich text editor with a powerful framework, a modular architecture, and modern features like collaborative editing. URL: <https://ckeditor.com/ckeditor-5/>.
- [6] WHO Disability Clarification. The World Health Organisation (WHO) launched the pioneering World report on disability in 2011 as a resource for policy makers, service providers, professionals and advocates for people with disability and their families. URL: <https://www.who.int/westernpacific/health-topics/disability>.
- [7] RIS - Bundes-Behindertengleichstellungsgesetz. Das Bundes-Behindertengleichstellungsgesetz (BGStG) bildet eines der gesetzlichen Grundlagen für das Behindertenrecht, dessen Ziel es ist, die Diskriminierung von Menschen mit Behinderungen zu beseitigen oder zu verhindern. URL: <https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=20004228>.
- [8] RIS - Behinderteneinstellungsgesetz. Das Behinderteneinstellungsgesetz (BEinstG) bildet eines der gesetzlichen Grundlagen für das

Behindertenrecht und regelt die Diskriminierung in der Arbeitswelt.
URL: <https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10008253>.

- [9] RIS - Bundesbehindertengesetz. Das Bundesbehindertengesetz (BBG) bildet eines der gesetzlichen Grundlagen des Behindertenrechts und regelt die Beratung, Betreuung und besondere Hilfe für Menschen mit Behinderungen. URL: <https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10008713>.
- [10] RIS - Web-Zugänglichkeits-Gesetz. Das Web-Zugänglichkeits-Gesetz (WZG) hat das Ziel Websites und mobile Anwendungen für alle Nutzerinnen und Nutzer, insbesondere für Menschen mit Behinderungen, barrierefrei zu gestalten. URL: <https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=20010727>.
- [11] Versionsverwaltung. Die Versionsverwaltung protokolliert Änderungen, die an einer Datei oder einer Reihe von Dateien über die Zeit hinweg getätigt wurden. URL: <https://www.it-talents.de/blog/it-talents/was-ist-eine-versionsverwaltung>.
- [12] What is Git. Git is a distributed version control system. Every dev has a working copy of the code and full change history on their local machine, by Linus Torvalds URL: <https://www.atlassian.com/git/tutorials/what-is-git>.
- [13] WCAG 2.1. Web Content Accessibility Guidelines (WCAG) 2.1 covers a wide range of recommendations for making Web content more accessible. URL: <https://www.w3.org/TR/WCAG21/>.
- [14] World Wide Web Consortium (W3C). The World Wide Web Consortium (W3C) is an international community where Member organizations, a full-time staff, and the public work together to develop Web standards. URL: <https://www.w3.org/Consortium/>.
- [15] Design Patterns. In software engineering, a design pattern is a general repeatable solution to a commonly occurring problem in software design. URL: https://sourcemaking.com/design_patterns.
- [16] Facotry Method Design Pattern. Factory Method Pattern defines an interface for creating an object, but lets subclasses decide which class to instantiate. URL: https://sourcemaking.com/design_patterns/factory_method.

Abbildungsverzeichnis

Tabellenverzeichnis

Anhang A

Arbeitsteilung

Anhang B

Protokolle

B.1 Protokoll vom 24. Juli 2020

Anwesende

- Prof. Thomas Stütz
- Halil Bahar
- Sonja Cao

Ort

Sprachkonferenz über Discord

Inhalt

Besprechung des Projektumfangs und der verwendeten Technologien

B.2 Protokoll vom 03. August 2020

Anwesende

- Prof. Thomas Stütz
- Halil Bahar
- Sonja Cao

Ort

Sprachkonferenz über Discord

Inhalt

Besprechung über die Erweiterung von Trix:

- Struktur des Projekts
- Verwendete Programmiersprachen sind JavaScript und TypeScript
- Bereitstellung der Barrierefreiheit

Auftrag bis zum nächsten Meeting

Erstellung einer Grafik für einen Überblick über die Schnittstellen und Beschreibung der notwendigen JavaScript Events.

B.3 Protokoll vom 11. August 2020

Anwesende

- Prof. Thomas Stütz
- Halil Bahar
- Sonja Cao

Ort

Sprachkonferenz über Discord

Inhalt

Besprechung der ersten schriftliche Ausarbeitung der Diplomarbeit:

- Bezeichnung der Kapitel
- Verbesserungsmöglichkeiten zur besseren Verständlichkeit der Arbeit

Auftrag bis zum nächsten Meeting

Verbesserung der aktuellen und Erstellung neuer Diagramme und Grafiken (Überblick über die Schnittstellen, DOM als Baumstruktur, Aussehen des Texteditors Trix) zum besseren Verständnis der Arbeit und Beginn der schriftlichen Ausarbeitung des Kapitel *Einleitung*.

B.4 Protokoll vom 31. August 2020

Anwesende

- Prof. Thomas Stütz
- Halil Bahar
- Sonja Cao

Ort

Sprachkonferenz über Discord

Inhalt

Besprechung der schriftlichen Ausarbeitung der Diplomarbeit:

- Korrigieren einzelner Rechtschreib- und Grammatikfehler
- Überlegungen über funktionale und nichtfunktionale Anforderungen

Auftrag bis zum nächsten Meeting

Beschreibung, warum der Texteditor in Fabasoft Produkten ersetzt werden muss, Verfassen kürzerer Sätze, Trennung zwischen Projektablauf und Produkt und Änderung der funktionalen und nichtfunktionalen Anforderungen.

B.5 Protokoll vom 07. September 2020

Anwesende

- Prof. Thomas Stütz
- Halil Bahar

Ort

Sprachkonferenz über Discord

Inhalt

Besprechung der schriftlichen Ausarbeitung der Diplomarbeit:

- Systemarchitektur mit und ohne der Erweiterung

Auftrag bis zum nächsten Meeting

Skizzieren der Systemarchitektur mit und ohne Erweiterung, Beschreibung der Zusammenhänge und Unterschiede und Verbessern der Beschreibung beim Kapitel "Funktionale Anforderungen".

B.6 Protokoll vom 21. September 2020

Anwesende

- Prof. Thomas Stütz
- Halil Bahar
- Sonja Cao

Ort

Sprachkonferenz über Discord

Inhalt

Besprechung der schriftlichen Ausarbeitung der Diplomarbeit:

- Kleine Grammatik- und Rechtschreibfehler
- Struktur der Kapitel
- Platzierung und Beschreibung der Systemarchitektur und des Texteditors

Auftrag bis zum nächsten Meeting

Ausbessern der Fehler, Zusätzliches Kapitel für erfüllte WCAG 2.1 Kriterien, Kapitel für Web Components

B.7 Protokoll vom 09. November 2020

Anwesende

- Prof. Thomas Stütz
- Halil Bahar
- Sonja Cao

Ort

Sprachkonferenz über Discord

Inhalt

Besprechung der schriftlichen Ausarbeitung der Diplomarbeit:

- Beschreibung der Begriffe Design Patterns und Factory Method Pattern
- Überarbeitung des Kapitels 3.5 JavaScript Events & EventListener

Auftrag bis zum nächsten Meeting

Ergänzung des Factory Method Patterns mit einem UML-Klassendiagramm