## 

i.A. Prof. Dr. Werner König

505 GUI - Programmieren Graphischer Oberflächen

2021, Leon Etienne, Issam Charef

Abhandlung zur Mehrsprachigkeit in der modernen

Informatik

*Schwerpunkt: Webentwicklung*

# Inhalt

[1 Inhalt 2](#_Toc72972953)

[2 Vorwort 3](#_Toc72972954)

[3 Problemstellung 3](#_Toc72972955)

[4 Zielsetzung 3](#_Toc72972956)

[5 Recherche 3](#_Toc72972957)

[5.1 Primärmethoden 3](#_Toc72972958)

[5.1.1 Dynamische Übersetzung 3](#_Toc72972959)

[5.1.2 Statische Übersetzung 3](#_Toc72972960)

[5.2 Ausgeschlossene Problemstellungen 4](#_Toc72972961)

[5.3 Vor- und Nachteile der versch. Methoden 4](#_Toc72972962)

[5.3.1 Dynamische Übersetzung 4](#_Toc72972963)

[5.3.2 Statische Übersetzung 4](#_Toc72972964)

[I. Implementation der Methoden 4](#_Toc72972965)

[II. Fazit 4](#_Toc72972966)

[III. Verweise 5](#_Toc72972967)

# Vorwort

Diese Ausarbeitung entstand im Rahmen des Kursmodules 505 – Programmierung Grafischer Benutzeroberflächen[[1]](#footnote-1) unter Kursführung durch Prof. Dr. Werner König, Hochschule Worms, University of Applied Sciences, in Form der zu erbringenden Prüfleistung. Ziel dieser Arbeit ist es, das Konzept, sowie Lösungsansätze der Mehrsprachigkeit in der Informatik, zu ermitteln, in der Praxis explorativ zu testen und diese Ergebnisse der Gesamtheit des Kursmodules zur Verfügung zu stellen.

# Problemstellung

Die Informatik ist ein internationales Geschehen. Sie steht über jeglicher Landesgrenze, jedoch bestehen immer noch Sprachbarrieren. Informatik ist heutzutage so allgegenwertig, dass sie fast schon ein Existenzkriterium für jedermann ist. Auch für diejenigen, die kein Englisch sprechen. So steht es an hoher Priorität niemanden zurückzulassen.

# Zielsetzung

Es soll gegeben sein, dass sämtliche linguistikabhängigen Ressourcen sprachübergreifend zur Verfügung stehen. Dies schließt hauptsächlich Text, aber auch Grafiken, sowie Videoquellen ein. Ebenso betroffen sind Hyperlinks, Sprachwiedergaben wie z.B. Podcasts und herunterladbare Dokumente.

# Recherche

In Anbetracht dessen, dass ein Teammitglied im Praxisverbund studiert und im Partnerunternehmen[[2]](#footnote-2) in der Webentwicklungsbranche tätig ist, bietet sich dem Team ein großes Spektrum an state-of-the-art Beispielen[[3]](#footnote-3), wie professionelle Anwendungsentwickler an die Themenstellung „Mehrsprachigkeit“ herangehen.

## Primärmethoden

Grundsätzlich unterteilen sich sämtliche Übersetzungsautomatismen in zwei Klassen: “Dynamische Übersetzung“, und „Statische Übersetzung“.

### Dynamische Übersetzung

Die dynamische Übersetzung zeichnet sich weitestgehend dadurch aus, dass sämtliche Inhalte maschinell übersetzt werden. Dies gescheit entweder über in der Applikation eingebettete Funktionsautomatismen, oder durch Dienste Dritter. Ein populärer Anbieter hierbei ist *Google[[4]](#footnote-4)* mit *Google Cloud Translation[[5]](#footnote-5)*.

Es ist nicht ausgeschlossen, dass auch Medien dynamisch übersetzt werden, jedoch ist das eine weitaus anspruchsvollere Problemstellung, die mit großen Parallelen zu maschinellem Lernen verbunden ist. Google bietet bereits diverse Anwendungen an, die Audioübersetzungen in Echtzeit liefern[[6]](#footnote-6) und sogar visuelle Medien übersetzen können[[7]](#footnote-7).

Die dynamische Übersetzung von (externen) Verlinkungen hingegen erweist sich als große Hürde. Hierfür müsste der angewandte Automatismus nicht nur die Grundessenz der Themenstellung der Webseite verstehen, sondern auch wissen, welche n-sprachige Quelle nun der nativ-sprachigen entsprechen würde.

### Statische Übersetzung

Bei der statischen Übersetzung ist jede Sprachvariante einer Quelle bereits im Voraus definiert. Hierfür gibt es verschiedene Möglichkeiten. Vor zehn Jahren (rel. 2021) war es noch allgegenwärtig, dass dieselbe Webseite in statischer Form mehrfach in verschiedenen Sprachvarianten gehostet wird. Heutzutage werden hierzu entweder\* Sprachdateien verwendet, in denen, meist in Form von Key-Value-Pairs (KVP)[[8]](#footnote-8), lokalisierte Textbruchstücke einem internen Identifier zugeordnet werden. Dort findet ebenfalls die Pflege statt. Diese Textbruchstücke können beliebiges darstellen, da man sie bedingungslos im Seitenquelltext einsetzen darf. So ist es nicht von Relevanz, ob ein KVP nun ein Wort übersetzt, eine angepasste URL zu einer Ressource, oder gar länderspezifische Elemente, wie z. B. ein Cookiebanner, enthält.

\*Oftmals, insbesondere bei Verwendung eines Content Management Systems (CMS)[[9]](#footnote-9), werden Sprachvarianten in einem Datenbanksystem gespeichert. Das Hauptmerkmal eines CMS ist, dass es eine Benutzeroberfläche bietet, um alle Inhalte, nicht nur, aber auch, die Sprachvarianten zu pflegen. Der große Vorteil hierbei ist, dass es eventuell Fehlermeldungen erheben kann, sollte eine Sprachvariante z. B. noch leer sein. Ebenfalls ist es nicht unüblich, dass CMS die Werte hinter des statischen Übersetzungsautomaten dynamisch genieren kann.

Unabhängig davon, ob statisch oder dynamisch übersetzt wird, muss es immer eine Fallback- Sprache oder Ressource geben. Sollte eine Sprachvariante nicht laden, oder das gefragte Element nicht gedeckt sein, sollte zumindest die Weltsprache angezeigt werden.

## Ausgeschlossene Problemstellungen

Einige Problemstellungen stellen sich als besonders herausfordernd dar. Diese Ansätze werden in dieser Ausarbeitung nicht näher dargelegt. Diese sind hauptsächlich, aber nicht ausschließlich:

1. Sprachartefakte, wie in z. B. Isländisch: Hier verändert sich die Schreibweise einzelner Wörter basierend auf dem Rest des Textes.[[10]](#footnote-10)
2. Dynamische Übersetzung von Medien.
3. Kombination von Right-To-Left (RTL) und Left-To-Right (LTR) orientierten Sprachsystemen.

(z. B. Erwähnung eines deutschen Namens in einem arabischen Text).

1. Zeitzonen.
2. Emoticons.

## Vor- und Nachteile der versch. Methoden

### Dynamische Übersetzung

Die dynamische Übersetzung ermöglicht es, insbesondere in großen Projekten, mit wenig Aufwand eine enorme Menge an Sprachen abzudecken.

Allerdings ist die Übersetzungsqualität oft mangelhaft und beschränkt sich darauf, dass Benutzer die Grundlegenden Informationen erhalten.

### Statische Übersetzung

Die Implementierung eines statischen Übersetzungsautomatismus ist weitaus unkomplizierter als ein die eines dynamischen. Im Kontrast dazu ist die Pflege eines großen, statisch übersetzten Projektes mit beträchtlichem Aufwand verbunden. Jede sprachabhängige Ressource, auch Text, muss sofort in jeder bereitgestellten Sprache angelegt werden. Änderungen müssen demgemäß in sämtlichen Sprachen erfolgen.

Dem Aufwand entsprechend kann die Übersetzungsqualität tadellos ausfallen. Im Idealfall beauftragt der\*die Redakteur\*in eine\*n Übersetzungsdienstleister\*in und lässt somit professionelle Sprachvarianten anfertigen.

# Implementation der Methoden

## Dynamische Übersetzung

Die dynamische Übersetzung kann in verschiedenen Methoden implementiert werden.

Es gibt verschiedenen Tools oder Frameworks, die diese anbieten. Im Fall der Webentwicklung kann der Google Translate API [Google Cloud Translation] sehr hilfreich sein, da es einfach implementierbar ist.

Eine komplette Erläuterung ist hier zu herunterladen.

Für Google Cloud Translation API oder auch andere dynamische Übersetzung API ist meistens ein PHP oder JavaScript Code, den in den eigenen Quellcode reingefügt wird und dann werden alle vorhandene Texten automatisch in alle anderen Sprachen übersetzt.

## Statische Übersetzung

### PHP

Die statische Übersetzung ist generell implantierbar, indem es alle Strings bzw. Texte in einer JSON – Datenbank – XML – Word oder auch Excel Datei schreibt und dann werden alle Texte manuell übersetzt. Für kleine Projekte ist die Übersetzung meistens vom Entwickler oder dem Projekinhaber zu machen aber für Größe Projekte wird sie über eine interne Dolmetscherabteilung oder auch externe für die Firma keine hat.

Eine komplette Erläuterung in der PHP-Sprache ist hier herunterzuladen.

### Java oder Kotlin

Bei Java und Kotlin die Geschickte sieht gleich aus außer dass die Textdateien, wo die Texte gespeichert sind, sind in einem XML zu finden, der vom IDE (Android Studio beispielerweise) generiert werden kann.

Eine komplette Erläuterung in der Android Studio ist hier herunterzuladen.

# Fazit

# Verweise

Autoren Wikipedias. *Wikipedia - Attribute-value pair.* 23. März 2021. https://en.wikipedia.org/wiki/Attribute%E2%80%93value\_pair (Zugriff am 27. Mai 2021).

—. *Wikipedia - Content management system.* 19. Mai 2021. https://en.wikipedia.org/wiki/Content\_management\_system (Zugriff am 27. Mai 2021).

Büro Medienagenten. *Büro Medienagenten - Startseite.* 2020. https://www.medienagenten.de/ (Zugriff am 26. Mai 2021).

—. *Referenzen der Medienagenten.* 2021. https://www.medienagenten.de/referenzen (Zugriff am 26. Mai 2020).

Google LLC. *Google.* 2021. www.google.de.

—. *Google Cloud Translation.* 2021. https://cloud.google.com/translate (Zugriff am 26. Mai 2021).

—. *Google Translate Help - Translate Images.* 2021. https://support.google.com/translate/answer/6142483?hl=en (Zugriff am 26. Mai 2021).

Prof. Dr. Herbert Thielen, Prof. Dr. Zdravko Bozakov. „Modulhandbuch-AnInf-2021.pdf.“ *Hochschule Worms.* 24. Mai 2021. https://www.hs-worms.de/fileadmin/media/fachbereiche/informatik/AInf/Modulhandbuch/Modulhandbuch-AnInf-2021s.pdf.

Scott, Thomas. *Internationalis(z)ing Code - Computerphile.* Webvideo, University of Nottingham: Computerphile, 2014.

Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.

1. (Prof. Dr. Herbert Thielen 2021) [↑](#footnote-ref-1)
2. (Büro Medienagenten 2020) [↑](#footnote-ref-2)
3. (Büro Medienagenten 2021) [↑](#footnote-ref-3)
4. (Google LLC 2021) [↑](#footnote-ref-4)
5. (Google LLC 2021) [↑](#footnote-ref-5)
6. (Google LLC 2021) [↑](#footnote-ref-6)
7. (Google LLC 2021) [↑](#footnote-ref-7)
8. (Autoren Wikipedias 2021) [↑](#footnote-ref-8)
9. (Autoren Wikipedias 2021) [↑](#footnote-ref-9)
10. (Scott 2014) [↑](#footnote-ref-10)