

Verteidigung Bachelorarbeit

Konzeption und prototypische Realisierung einer Webanwendung zur Simulation der Abdeckung von Richtantennen auf einer interaktiven Karte

Michael Lux

22. April 2021

Gliederung

Nutzen und Anwendungen

Problem- und Zielstellung

Erarbeitung der Anforderungen

Konzept

Implementierung

Fazit

Was? - Richtantennen und deren Abdeckung



Abbildung: Installation einer Richtantenne auf einem Dach

Quelle: <https://wiki.freifunk-franken.de/w/Richtfunk>



Abbildung: Abdeckung einer Richtantenne aus im Praxisprojekt entwickelter Simulationsanwendung

Warum? - Planung von Richtfunkstrecken

innerhalb freier Funknetze & Drahtlos-Internet-Dienstanbieter (WISP)



Abbildung: Initiativen für freie Funknetze: Freifunk in Deutschland und FunkFeuer in Österreich

<https://freifunk.net>

<https://funkfeuer.at>

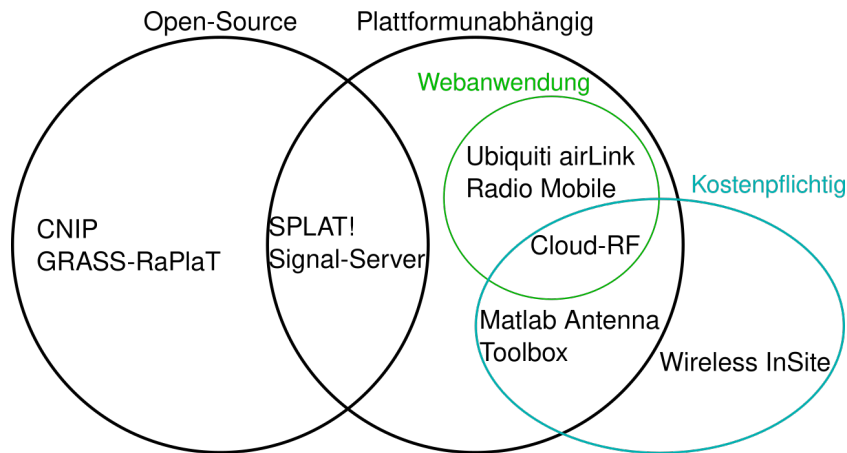


Abbildung: nicht kommerzielle WISP: Reudnetz w.V. und Westnetz w.V. aus Leipzig

<https://reudnetz.org>

<http://westnetz.org>

Wie? - Überblick verwendbarer Anwendungen



Problemstellung

Anwendungen weisen folgende Nachteile auf:

- ▶ Closed-Source
- ▶ Plattformabhängigkeit
- ▶ erschwerte Bedienbarkeit

Zielstellung

Konzeption und prototypische Realisierung einer Webanwendung zur Simulation der Abdeckung von Richtantennen auf einer interaktiven Karte

- ▶ Open-Source
- ▶ Plattformunabhängig
- ▶ Gute Bedienbarkeit

Mithilfe folgender Technologien:

- ▶ Leaflet¹
- ▶ Signal-Server
- ▶ Spring-Framework²

¹<https://leafletjs.com/>

²<https://spring.io/>

Erarbeitung der Anforderungen

Anforderungen klären



Konzept erarbeiten



Implementierung

Vorab zu klärende Fragen

1. Wie wird die Antennenabdeckung berechnet?
2. Wie kann die Simulation innerhalb einer Anwendung gestaltet werden?

1. fachspezifisches Wissen sammeln

Verwendung von Kanalmodellen

- ▶ berechnen Signalverluste im Funkkanal
- ▶ nach Phillips et al. [2] kategorisierbar (u.a. theoretische, Grund-,Gelände-, ergänzende und Vielstrahl-Modelle)
- ▶ benötigen Informationen über

Antennen (Antenneneigenschaften)

- ▶ Sendeleistung, Antennengewinn, Richtcharakteristik, Frequenz, Position und Ausrichtung, Polarisation, ...

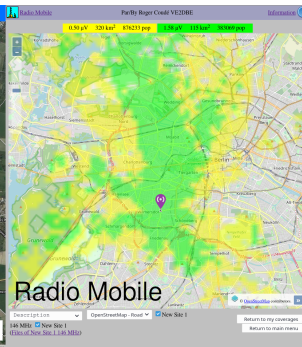
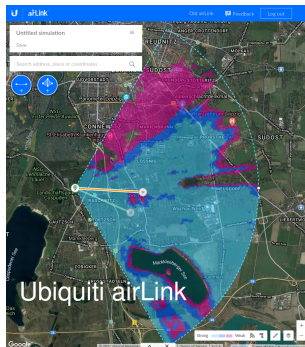
Ausbreitungsraum (Eigenschaften des Raumes)

- ▶ Höhendaten verschiedener Modelle benutzbar zur Beschreibung des Geländes, Klima-Zone, ...

2. Vergleich einzelner Anwendungen

vorher **jetzt**

- um Überblick verwendbarer Anwendungen zu erhalten
- um Anforderungen an eigene Anwendung zu ermitteln
- Erhalt erster Ideen für das Konzept



Anforderungsanalyse

Festlegung der Anforderungen

Funktional

- ▶ Eintragen, Verschieben und Löschen von Antennenstandorten
- ▶ Veränderung Antenneneigenschaften
- ▶ Auswahl des Kanalmodells
- ▶ Darstellung Abdeckung an ausgewählten Antennenstandorten
- ▶ Export Antennenstandorte mit -Eigenschaften

Nicht-funktional (Begriffe nach DIN/ISO 25010)

- ▶ Gute Bedienbarkeit
- ▶ Erweiterbarkeit
- ▶ Genauigkeit
- ▶ Hilfsbereitschaft
- ▶ Installierbarkeit

Entwurf der Architektur - Schichtenmodell

Entworfen nach Darstellung durch Starke [3]

Anwendung ▶ Webbrowser



Darstellung ▶ Webserver, Web-API-Controller

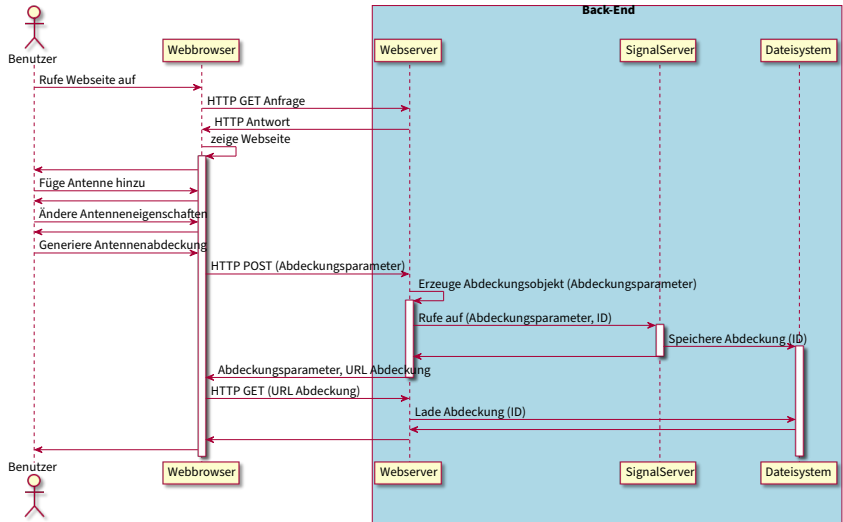


Modell ▶ Abdeckungs-Objekt



Infrastruktur ▶ Persistente Daten (Abdeckungsbilder, Template, Höhendaten, Antennendiagramme), Signal-Server

Entwurf der Architektur - Laufzeitsicht



Entwurf der Webseitenoberfläche

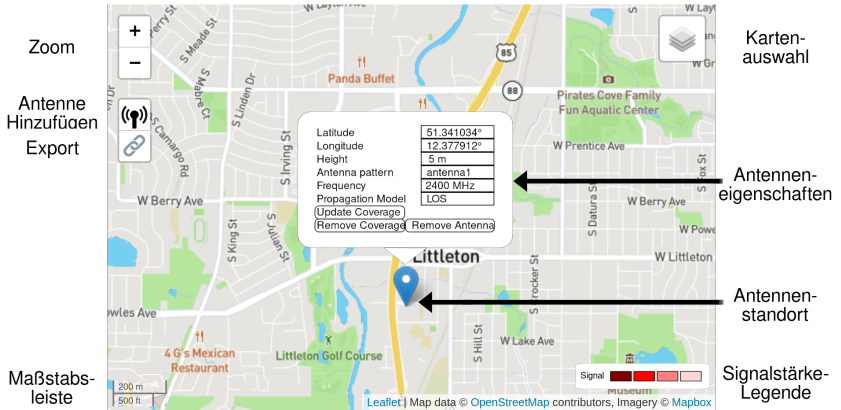


Abbildung: GUI-Prototyp mit Kartenansicht und Bedienelementen

Entwurf der Web-API

Entworfen nach dem REST-Prinzip von Fielding [1] und Beschreibung dessen von Starke [3]

Mapping	Controller	Funktion
<i>GET, “/”</i>	<i>Home</i>	<i>Erzeugung der Webseitenansicht</i>
<i>POST, “/coverage”</i>	<i>Coverage</i>	<i>Erzeugung eines Abdeckungsobjektes mit Generierung der Abdeckung</i>
<i>GET, “/coverage/ID”</i>	<i>Coverage</i>	<i>Rückgabe der geforderten Abdeckung</i>

Tabelle: Endpunkte der Webanwendung

Aufbau des Projektes

Spring-Projekt

- ▶ nach Ausführung durch Craig [4] erstellt
- ▶ erstellt mit *Spring initializr*³ -> automatische Einbindung der Abhängigkeiten
- ▶ Maven⁴ als Build-Tool -> Vorkonfigurationen, einfacher Build-Prozess durch Wrapper-Script
- ▶ Plattformunabhängig durch JVM-Umgebung

³<https://start.spring.io/>

⁴<https://maven.apache.org/>

Back-End-Komponenten

Verwendete Abhängigkeiten

- ▶ spring-boot-starter-web: Spring MVC (für Web-API) und Tomcat-Webserver
- ▶ spring-boot-starter-thymeleaf: Generierung Webseite
- ▶ spring-boot-starter-test: Testklasse für Schnittstelle zu Signal-Server
- ▶ springdoc-openapi-ui: Dokumentation der Web-API

Web-API

- ▶ Abdeckungs-Objekt gibt übergebene Abdeckungsparameter als Antwort zurück mit Status der Erstellung
- ▶ Validierung übergebener Parameter
- ▶ Daten-Sicherheit: Generierung eines zufälligen Hashwertes für ID einer Abdeckung

Generierung der Webseite



Thymeleaf⁵

- ▶ Java-Template-Engine
- ▶ erzeugt View (HTML-Seite) aus View (HTML)-Template
- ▶ fügt Liste von Kanalmodellen und Antennendiagramme in HTML-Template ein
- ▶ Erweiterung der Listen ohne Anpassung des Templates möglich

⁵<https://thymeleaf.org/>

Einrichtung von Signal-Server

Verzeichnisstruktur und Programm unverändert

Verwendete Höhendaten: Shuttle Radar Topography
Mission-Höhendaten mit $\sim 90\text{m}$ Auflösung

- ▶ frei verfügbar⁶ und verwendbar
- ▶ Abdeckung weiter Teile der Erdoberfläche
- ▶ im Vergleich zu hochauflösenden Höhendaten kurze
Berechnungszeit der Abdeckung einer Antenne

Antennendiagrammdaten von in freien Funknetzen häufig
verwendeten Richtantennen benutzt

⁶etwa unter <https://earthexplorer.usgs.gov/>

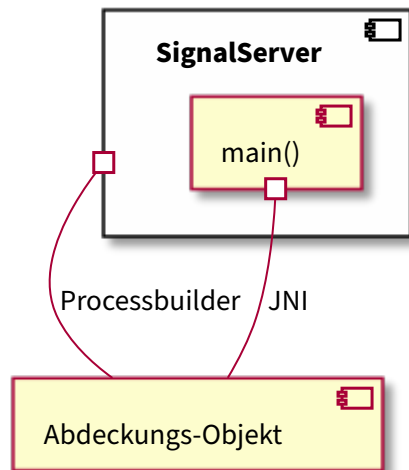
Kommunikation zwischen Webanwendung und Signal-Server

Benutzte API: Processbuilder

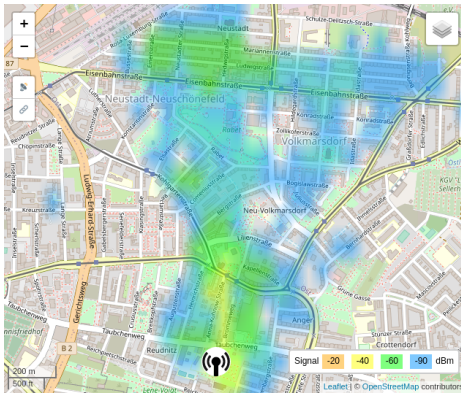
Verglichen mit Java Native Interface (JNI):

- ▶ Robust: Kapselung in eigenem Prozess
- ▶ Fehlerbehandlung mithilfe Ausgaben des Prozesses
- ▶ Aufruf einzelner Funktionen nicht möglich

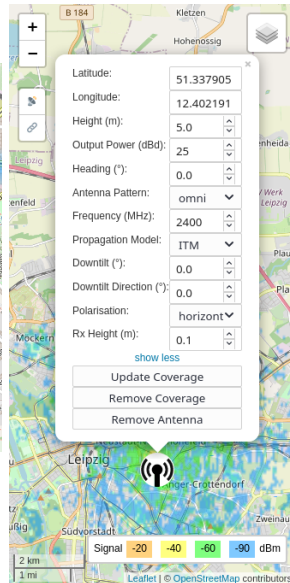
weitere Betrachtung durch Yew und Oxley [5]



Bedienelemente der Webseite



Ansicht der Webseite auf verschiedenen Endgeräten



Erreichte Ziele

- ▶ plattformunabhängige Open-Source-Alternative erschaffen
- ▶ Umsetzung aller funktionalen Anforderungen
 - ▶ **Signal-Server**: breite Auswahl konfigurierbarer Antenneneigenschaften, Abdeckungsparametern & Kanalmodellen
 - ▶ **Leaflet**: Auswahl und Darstellung Antennenstandort und -Abdeckung auf interaktiver Kartenansicht
 - ▶ **JSZip**: Export Antennenstandorte mit -Eigenschaften github.com/Stuk/jszip
- ▶ prototypischer Nachweis der Realisierbarkeit des Konzeptes mithilfe der Technologien gelungen

Umsetzung der Qualitätsziele

Gute Bedienbarkeit:

- ▶ GUI passt sich Anzeigebereich an
- ▶ alle Schritte zur Simulation auf Kartenansicht
- ▶ Bezug Höhen-/Antennendiagramm-Daten durch Nutzer entfällt

Erweiterbarkeit:

- ▶ durch weitere Spring-Frameworks, Leaflet-Plugins und Kanalmodelle
- ▶ Modularität des Schichtenmodells

Genauigkeit:

- ▶ getestete Implementationen etablierter und überprüfter Kanalmodelle

Hilfsbereitschaft:

- ▶ Infoboxen in GUI, Dokumentation für Bau, Einsatz und Quellcode

Installierbarkeit:

- ▶ vereinfacht durch Maven-Wrapper

Ausblick

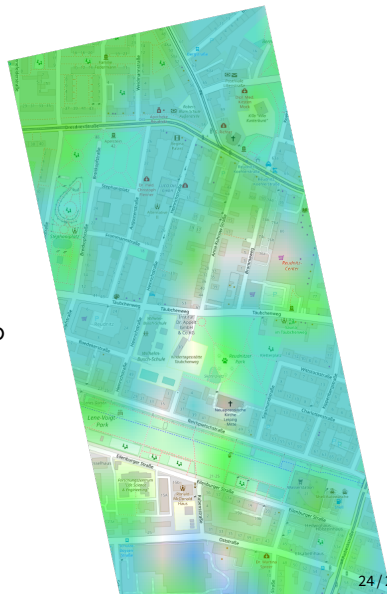
- ▶ Verwendung in freien Funknetzen
- ▶ Erweiterungen und Anpassungen



RadioMap

<https://github.com/lmux/RadioMap>

<http://radiomap.lmux.de/>



Rückblick auf die Arbeit

- ▶ alternative Herangehensweisen und Fokus denkbar
- ▶ Beitrag zur Unterstützung freier Funknetze und nicht-kommerzieller ISPs

FAKULTÄT INFORMATIK UND MEDIEN
DER HOCHSCHULE FÜR TECHNIK, WIRTSCHAFT UND KULTUR LEIPZIG

HTWK

Bachelorstudiengang Informatik

Bachelorarbeit

Zur Erlangung des akademischen Grades

Bachelor of Science (B.Sc.)

**Konzeption und prototypische
Realisierung einer Webanwendung zur
Simulation der Abdeckung von
Richtantennen
auf einer interaktiven Karte**

Eingereicht von: Michael Lux
Matrikelnummer: 70769

Erstprüfer: Prof. Dr. Jean-Alexander Müller

Zweitprüfer: Dr. Toni Tontchev

Abgabetermin: 18. März 2021

Referenzen

- [1] Roy Thomas Fielding. “Architectural Styles and the Design of Network-based Software Architectures – Dissertation”. Diss. University of California, Irvine, 2000, S. 76.
- [2] Caleb Phillips, Douglas Sicker und Dirk Grunwald. “A Survey of Wireless Path Loss Prediction and Coverage Mapping Methods”. In: *Communications Surveys & Tutorials, IEEE* 15 (Jan. 2013), S. 255–270. DOI: 10.1109/SURV.2012.022412.00172.
- [3] Gernot Starke. *Effektive Softwarearchitekturen: Ein praktischer Leitfaden*. 9. Auflage. Carl Hanser Verlag GmbH Co KG, Aug. 2020.
- [4] Craig Walls. *Spring im Einsatz*. 3. Auflage. Carl Hanser Verlag GmbH Co KG, 2020.
- [5] Kwang Hooi Yew und Alan Oxley. “JNI Fault Tolerance Using Java ProcessBuilder”. In: *International Journal of Information Technology, Electronics and Communication* 9 (Jan. 2014), S. 20–24.

Behandelte Anwendungen

Ubiquiti airLink <https://link.ui.com/>

Radio Mobile <https://ve2dbe.com/english1.html>

Cloud-RF <https://cloudrf.com/>

CNIP <https://github.com/Telecominfraproject/OpenCellular/tree/cnip/software/cnip>

GRASS-RaPlaT http://www-e6.ijs.si/RaPlaT/GRASS-RaPlaT_main_page

Matlab Antenna Toolbox <https://mathworks.com/products/antenna.html>

Wireless InSite

<https://remcom.com/wireless-insite-em-propagation-software/>

SPLAT! <https://qsl.net/kd2bd/splat.html>

Signal-Server (Fork) <https://github.com/N90ZB/Signal-Server>