

## Bachelor- und Masterarbeiten

Fachrichtungen Mathematik / Informatik / Wirtschaftsinformatik

Die Navimatix GmbH ist ein Softwareentwicklungsunternehmen mit Sitz in Jena und seit über 10 Jahren erfolgreich am Markt tätig ist. Der Fokus liegt auf der Entwicklung individuell angepasster Softwarelösungen für kleine und große Unternehmen aus verschiedenen Bereichen, wie zum Beispiel der Automobilbranche, Transport & Logistik und dem Gesundheitswesen.

Wir suchen engagierte Verstärkung und bieten Bachelor- / Masterarbeiten zu folgenden Themen:

### A1 - Einheitliche Build-Infrastruktur für heterogene Software-Architekturen

Innerhalb der Navimatix GmbH wird eine Vielzahl von Programmiersprachen und Software-Technologien eingesetzt. Diese haben unterschiedliche Anforderungen an automatische Build- und Deployment-Systeme. In dieser Arbeit sollen die Anforderungen der verschiedenen Teams gesammelt und kategorisiert werden. Danach sollen verfügbare Build-Systeme auf die Erfüllung dieser Anforderungen geprüft und bewertet werden.

**Technologien/ Tools:** Continuous Integration und Continuous Deployment, C/C++, Java, JavaScript, Node.js, Docker

**Erforderliche Kenntnisse:** Dokumentation von Software-Anforderungen, grundlegende Kenntnisse über Build-Prozesse in einer der erwähnten Sprachen

### A2 - Mehrmandantenfähigkeit für ein Angular 7 Frontend

Für ein Angular-basiertes Frontend ist ein Konzept zu entwickeln, wie es mit möglichst wenig Aufwand an das Corporate Design verschiedener Kunden („Mandanten“) angepasst werden kann. Der Nutzer soll also ein an sein Unternehmen angepasstes Design und angepasste Funktionen erhalten, obwohl das Frontend auch in anderen Unternehmen eingesetzt wird. Im Idealfall wird das Design bei Anmeldung des Nutzers dynamisch angepasst.

**Technologien/ Tools:** Angular, Typescript, CSS, REST APIs

**Erforderliche Kenntnisse:** Gute Angular-Kenntnisse notwendig

### A3 - Testkonzepte für webbasierte Client-Server-Anwendungen

Client-Server-Anwendungen nutzen und implementieren viele Schnittstellen. Bei Microservice-Anwendungen kommen noch zusätzliche Schnittstellen hinzu. Daher sind eine Vielzahl von Unit-

Integrations- und Systemtests möglich. Ziel der Arbeit soll sein, aus dieser Testvielfalt die relevanten/kritischen Tests zu identifizieren. Im zweiten Schritt müssen dann entsprechende Testtechnologien gefunden werden, die es ermöglichen die als relevant eingestuften Tests automatisiert durchzuführen.

**Technologien/ Tools:** Spring-Boot-Microservices, JUnit, Rest Assured, H2 In-Memory-Datenbank

**Erforderliche Kenntnisse:** Erfahrung mit Software Tests und Testkonzepten, Microservices oder einer anderen web-basierten Architektur

## A4 - Erstellen und Konfigurieren von Repositories für Node.js-Pakete und Docker-Images

Um automatisch auf Software-Bibliotheken und andere Software-Artefakte in einer bestimmten Version zugreifen zu können, werden sogenannte Repositories verwendet. Soll ein solches Repository nur innerhalb eines lokalen Netzwerks zugreifbar sein, muss eine entsprechende Server-Anwendung lokal installiert werden. Hier soll für 2 Typen von Repositories eine solche Installation durchgeführt und überprüft werden, welche Konfiguration dieser Repository-Anwendungen für den Nutzer am komfortabelsten ist.

**Technologien/ Tools:** Docker, node.js, Linux-Administration

**Erforderliche Kenntnisse:** Linux-Terminal/bash, Konfiguration von Linux-Daemons, Docker-Kenntnisse hilfreich

## A5 - Automatisiertes Testen von Web-Frontends

Für das Testen von Angular Web Applikationen gibt es eine Vielzahl von Frameworks wie z.B. Selenium und Poltergeist. In dieser Arbeit soll untersucht werden, welche Vor- und Nachteile die Frameworks in Bezug auf die Integration in eine Continuous Integration Pipeline haben. Wenn möglich, wird eine Continuous Integration Pipeline mit dem am besten geeigneten Framework erstellt und dokumentiert.

**Technologien/ Tools:** Angular, automatisierte Browser Tests, Continuous Integration mit GitLab CI

**Erforderliche Kenntnisse:** Angular, Erfahrung mit Software-Tests, idealerweise mit Frontend-Tests

## A6 - Data Mining auf OSM Kartendaten: Vorberechnung einer Datenstruktur für Administrativen und deren Zuordnungen

Die Kartendaten von OpenStreetMap (OSM) haben sich in den letzten Jahren sehr stark weiterentwickelt und haben bereits jetzt in einigen Bereichen eine höhere Informationsdichte als die Daten von kommerziellen Anbietern. Besonders die Anzahl von Straßen, Fuß- und Fahrradwegen ist enorm und in ihrer Gesamtheit einzigartig.

Dem steht leider gegenüber, dass beim Kartieren der OSM-Daten häufig starkes Augenmerk auf die Visualisierung der Daten gelegt wird, nicht jedoch deren Strukturierung und die Beziehungen zwischen den Kartenelementen. Eine Verarbeitung von OSM-Daten mit Algorithmen ist daher oft aufwendig und komplex.

Speziell bei den Administrativen Strukturen existieren aktuell (und sollen anscheinend auch zukünftig nicht verfügbar gemacht werden) keine direkten Zuordnungen untereinander. Informationen der Zugehörigkeit von Stadtteilen und Städten zu deren Landkreisen, Bundesländern oder Ländern können den Daten nicht direkt entnommen werden, sondern müssen aufgrund der geografischen Lage und anderer Parameter berechnet werden. Durch die verwendbaren geometrischen Algorithmen und die hohe Anzahl der Administrativen in Europa oder dem Weltweiten Datenbestand bei OSM, ist diese Berechnung komplex und sehr zeitaufwendig.

Ziel dieser Arbeit ist es, einen möglichst effizienten Weg dieser Berechnung zu planen und komplett oder in Teilen zu implementieren. Darüber hinaus sollen die Ergebnisse in einer ebenso zu entwerfenden Datenstruktur (z.B.: Datenbank) so gespeichert werden, dass Änderungen an den OSM-Daten selbst möglichst nicht zu einer vollständigen Neuberechnung der gesamten Zuordnungen führen. Die Ergebnisdaten sollen also inkrementell aktualisierbar sein.

**Technologien/ Tools:** OpenStreetMap, Navimatix Kartentechnik, eventuell GIS, geografische Daten, Kartenformate

**Erforderliche Kenntnisse:** Geoinformatik, Informatik, Programmierung, Softwaretechnik, Algorithmen und Datenstrukturen

## A7 - Data Mining auf OSM Kartendaten: Effektive Zuordnung von Straßen zu Administrativen

Die Kartendaten von OpenStreetMap (OSM) haben sich in den letzten Jahren sehr stark weiterentwickelt und haben bereits jetzt in einigen Bereichen eine höhere Informationsdichte als die Daten von kommerziellen Anbietern. Besonders die Anzahl von Straßen, Fuß- und Fahrradwegen ist enorm und in ihrer Gesamtheit einzigartig.

Dem steht leider gegenüber, dass beim Kartieren der OSM-Daten häufig starkes Augenmerk auf die Visualisierung der Daten gelegt wird, nicht jedoch deren Strukturierung und die Beziehungen zwischen den Kartenelementen. Eine Verarbeitung von OSM-Daten mit Algorithmen ist daher oft aufwendig und komplex.

Die Zuordnung von Straßen zu Administrativen, etwa Stadtteilen und Städten, ist nicht explizit in den Daten vorhanden und muss aufwendig berechnet werden. Dieses geschieht aufgrund der geografischen Lage und anderer Parameter. Durch die verwendbaren geometrischen Algorithmen und die hohe Anzahl der Straßen und Administrativen in Europa oder dem Weltweiten Datenbestand bei OSM, ist diese Berechnung komplex und sehr zeitaufwendig.

Ziel dieser Arbeit ist es, einen effizienten Weg dieser Berechnung zu planen und komplett oder in Teilen zu implementieren. Darüber hinaus sollen die Ergebnisse in einer ebenso zu entwerfenden Datenstruktur (z.B.: Datenbank) so gespeichert werden, dass Änderungen an den OSM-Daten selbst möglichst nicht zu einer vollständigen Neuberechnung der gesamten Zuordnungen führen. Die Ergebnisdaten sollen also inkrementell aktualisierbar sein und für die Navimatix-interne Kartenerstellung verfügbar gemacht werden.

**Technologien/ Tools:** OpenStreetMap, Navimatix Kartentechnik, eventuell GIS, geografische Daten, Kartenformate

**Erforderliche Kenntnisse:** Geoinformatik, Informatik, Programmierung, Softwaretechnik, Algorithmen und Datenstrukturen

## A8 - Algorithmen/Datenstrukturen: Entwurf eines Algorithmus für die Zuordnung von geografischen Primitiven, Punkte, Linien und Flächen in einen Geo-Index (Gridfile)

Für einen schnellen Zugriff auf Kartendaten benötigen Kartendienste spezielle Datenstrukturen, die einen entsprechend schnelles Lesen von einzelnen oder mehreren Kartenelementen erlauben. Diese als Geo-Index bekannten Datenstrukturen sind sehr unterschiedlich aufgebaut, es existieren viele verschiedene Varianten und Konzepte.

Ein Konzept, Grid-File, steht hier im Vordergrund und soll im Rahmen dieser Arbeit betrachtet werden. Dabei werden alle Kartenelemente in regelmäßige Zellen, sog. Kacheln, geografisch einsortiert und stehen so in einer Nähebeziehung. Nahe beieinander liegende Elemente befinden sich in der gleichen, oder in einer Nachbarkachel. Der schnelle Zugriff auf die Kacheln und damit die Kartenelemente ist hier nicht von Bedeutung, es geht lediglich um das Einsortieren der Elemente in Kacheln, also um das Aufbauen des Geo-Index.

Es wird die Herausforderung sein, entsprechende Algorithmen zu entwerfen, die eine schnelle Einsortierung von Punkten, Linien und Flächen in den Index ermöglichen. Insbesondere, da es in den zu nutzenden Kartendaten Flächen großen Ausmaßes (etwa die Größe eines Landes) mit mehreren 10tausend Punkten gibt, die in entsprechend viele Kacheln einsortiert werden müssen.

Ziel dieser Arbeit ist es, oben genannte Algorithmen zu planen und komplett oder in Teilen umzusetzen. Eine Bewertung der realen Laufzeit der Algorithmen wird ebenso angestrebt.

**Technologien/ Tools:** Navimatix Kartentechnik, GIS, geografische Daten, Kartenformate

**Erforderliche Kenntnisse:** Kartendaten allgemein, Geoinformatik, Informatik, Programmierung, Softwaretechnik, Algorithmen und Datenstrukturen, Algorithmische Geometrie

## A9 - Entwurf einer Datenstruktur zur vollständigen Integration von geografischen Elementen in Kacheln für ein gekacheltes Kartenformat digitaler Karten

Für einen schnellen Zugriff auf Kartendaten benötigen Kartendienste spezielle Datenstrukturen, die einen entsprechend schnelles Lesen von einzelnen oder mehreren Kartenelementen erlauben. Diese als Geo-Index bekannten Datenstrukturen sind sehr unterschiedlich aufgebaut, es existieren viele verschiedene Varianten und Konzepte.

Ein Konzept, Grid-File, steht hier im Vordergrund und soll im Rahmen dieser Arbeit betrachtet werden. Dabei werden alle Kartenelemente in regelmäßige Zellen, sog. Kacheln, geografisch einsortiert und stehen so in einer Nähebeziehung. Nahe beieinander liegende Elemente befinden sich in der gleichen, oder in einer Nachbarkachel.

Der schnelle Zugriff auf die Kacheln oder die Entwicklung eines effizienten Algorithmus für die Zuordnung der Elemente steht hier nicht im Vordergrund. Vielmehr handelt es sich um die Planung und Implementierung einer Datenstruktur zur Aufnahme aller Kartenelemente in die Kachelstruktur. Dabei sollen alle Kartenelemente samt ihrer Eigenschaften und Geometrien direkt in die Kacheln übernommen werden. Diese so geclusterten Daten sollen einen schnellstmöglichen Geo-Index gewährleisten, welcher beim Suchen mit einem Minimum an IO-Operationen auskommt.

In dieser Arbeit sind Konzepte zu entwickeln, wie etwa welche Informationen welcher Kartenelemente sinnvollerweise in welche Kacheln integriert werden. Auch die Definition der inneren Datenstruktur einer Kachel wird ein Hauptbestandteil sein.

**Technologien/ Tools:** Navimatix Kartentechnik, GIS, geografische Daten, Kartenformate

**Erforderliche Kenntnisse:** Kartendaten allgemein, Geoinformatik, Informatik, Programmierung, Softwaretechnik, Algorithmen und Datenstrukturen, Algorithmische Geometrie

## A10 - Algorithmus- Anpassung und Implementierung Routenberechnung „Highway Hierarchien“ für Navimatix Kartenformat und Kartentechnik

Eines der zentralen Themen im Bereich digitale Karten ist das Thema Routenberechnung. Dabei werden auf den gegebenen digitalen Karten kürzeste, schnellste oder wirtschaftlichste Wege von einem Startpunkt zu einem Ziel berechnet. Es existieren bereits gut bekannte Algorithmen, um diese Wege zu berechnen, etwa der Algorithmus von Dijkstra oder der A\*-Algorithmus. Konzeptbedingt sind die Optimierungsmöglichkeiten dieser klassischen Ansätze eingeschränkt. Die Dijkstra-artigen Algorithmen benötigen Informationen über sehr viele einzelne Straßenelemente und haben daher mitunter hohe Laufzeiten für sehr große Straßennetze.

Filtert man Straßen von geringem Interesse aus, so kann man eine deutliche Laufzeitoptimierung erreichen. Berechnet man sogar bestimmte Routen komplett voraus, speichert diese mit den Kartendaten und verbindet lediglich kurze und vorberechnete Routen miteinander, so sind noch weitere Optimierungsmöglichkeiten offen.

Ziel dieser Arbeit ist es, eine spezielle Art von Algorithmen mit zugehöriger Datenstruktur in diesem Bereich, sog. „Highway Hierarchies“, mit der Navimatix Kartentechnik zu verbinden und zu implementieren.

**Technologien/ Tools:** Navimatix Kartentechnik, GIS, geografische Daten, Kartenformate

**Erforderliche Kenntnisse:** Kartendaten allgemein, Geoinformatik, Informatik, Programmierung, Softwaretechnik, Algorithmen und Datenstrukturen, Algorithmische Geometrie

## A11 - Planung und Erstellung eines objektorientierten Interface Generators für DLLs und so. Definition in Meta Sprache

Häufig stehen Softwareentwickler vor der Aufgabe, eine zu erstellende Software in einzelne Module aufzuteilen, welche getrennt voneinander erstellt werden können und auf binärer Ebene miteinander interagieren. Die Möglichkeit dazu wird oft schon vom Betriebssystem bereitgestellt, etwa die DLL's unter Windows oder SharedObjects (.so) für Linux.

Für maximale Kompatibilität existieren hier jedoch nur einfache Konzepte, eine flache Funktionsliste ohne objektorientierte Elemente und nur einfache Datentypen zur Parameterübergabe zwischen den Modulen. Auch mit Speicher- und Ressourcenverwaltung müssen sich Entwickler im Normalfall immer wieder neu beschäftigen.

Ziel dieser Arbeit ist es, einen Generator für Schnittstellen zu erstellen, der diese Problematik entschärft. In einem Metamodell (eigene Sprache, Datenstruktur oder existierende IDL) soll eine Schnittstelle zwischen Modulen, auch objektorientiert, definiert werden können. Alle Funktionsaufrufe und Objekte mit Methoden und Attributen, sowie die genaue Parametrisierung aller Aufrufe sollen hier enthalten sein. Aus diesem Metamodell sollen sich dann beide Schnittstellenteile der zwei beteiligten Module generieren lassen. Beide generierten Schnittstellenseiten sollen bei Bedarf objektorientiert sein können und die Übergabe komplexer Datentypen erlauben. Komplexe Datentypen müssen vom Generator entsprechend so umgesetzt werden, dass Compiler- und Programmiersprachengrenzen überwunden werden.

Bei Änderungen der Schnittstelle selbst kann entsprechend neu generiert werden, es müssen dann nur die an der Schnittstelle getätigten Änderungen in der Implementierung angepasst werden.

Als Abgrenzung zu ähnlichen, existierenden Ansätzen wie etwa SWIG oder Swagger, sei eine Generierung von einfachen, menschenverständlichen Wrappern und ein Bezug auf Bibliotheken, DLL und so (kein http-/Netzwerkbezug wie bei Swagger), definiert.

**Technologien/ Tools:** DLL's, so, Zu unterstützende Sprachen: C/C++, C# (mit Interop), Java (JNI, JNA), Delphi/FPC, sonstige...

**Erforderliche Kenntnisse:** Softwaretechnik, Objektorientierte Programmierung und Analyse, Interoperabilität (Programmiersprachen), Parsen



## A12 - Marktanalyse: Mobility Services und ecall

Mobility Services wie Emergency Call, Information Call (Concierge), Stolen Vehicle Tracking, Breakdown Call und andere Dienste durchdringen mehr und mehr den Markt, insbesondere durch die großen Fahrzeughersteller selbst, welche ihre Neufahrzeuge ab Werk mit eben diesen Diensten ausrüsten. Ältere Fahrzeuge und Spezialfahrzeuge kleinerer Hersteller benötigen Nachrüstlösungen, damit deren Besitzer an diesen Diensten ebenso teilnehmen können. Dafür gibt es bereits günstige Hardwarelösungen im Zusammenhang mit Smartphones, welche hauptsächlich Gegenstand dieser Analyse sein sollen. Besonderes Augenmerk soll auf die Erschließung weiterer Geschäftsfelder, sowie der zugehörigen Zielgruppen gelegt werden. Daher ergeben sich die folgenden Aufgaben:

- Potentielle neue Produkte definieren und auf deren Markttauglichkeit testen
- Analyse der Randbedingungen, etwa gesetzlichen Regularien, Markteintrittsproblematiken und Multiplikatoren
- Definition von Zielgruppen und Erarbeiten von Marketingstrategien für diese

**Ergebnisse:** Das Ergebnis soll die Marktanalyse in Form eines Dokumentes sein.

**Kenntnisse:** Keine speziellen IT-Kenntnisse erforderlich, Betriebswirtschaft, Marketing

Bei Navimatix arbeiten Sie als Student mit Ihren Kollegen auf Augenhöhe. Wir arbeiten in dynamischen, interdisziplinären Teams und bieten Ihnen individuelle Entwicklungsperspektiven sowie Flexibilität bei der Gestalt Ihrer Arbeit.

**Interessiert?** Dann freuen wir uns auf Ihre Bewerbungsunterlagen: [jobs@navimatix.de](mailto:jobs@navimatix.de)