

Fachpraktikum WS2020/2021

Erstellung der Steuersoftware für einen Roboter zur autonomen Navigation durch einen Hindernisparkour und Kartografierung des selbigen

Team 3

Software-Systemarchitektur

Prüfer: Prof. Dr.-Ing. Dr. h. c. Michael Weyrich
Betreuer: Hannes Vietz

Start: 08.12.2020

Abgabe: 15.12.2020



Universität Stuttgart
Institut für Automatisierungstechnik
und Softwaresysteme

Dokument Versionsverwaltung

Version	Autor	QS	Datum	Status	Änderungen
0.1	WaRe	MiSt	14.12.2020	durch QS	Erstellung

0 Inhaltsverzeichnis

0 INHALTSVERZEICHNIS.....	3
1 UMGEBUNGS- UND RANDBEDINGUNGEN.....	4
2 GRUNDLEGENDE ENTWURFSENTSCHEIDUNGEN.....	4
3 DIAGRAMM DER SOFTWARE-SYSTEMARCHITEKTUR.....	4
4 SOFTWAREKOMPONENTEN.....	4
5 SCHNITTSTELLEDEFINITIONEN.....	5
5.1 Schnittstelle Controller.....	5
5.2 Schnittstelle Visualisierung.....	5

1 Umgebungs- und Randbedingungen

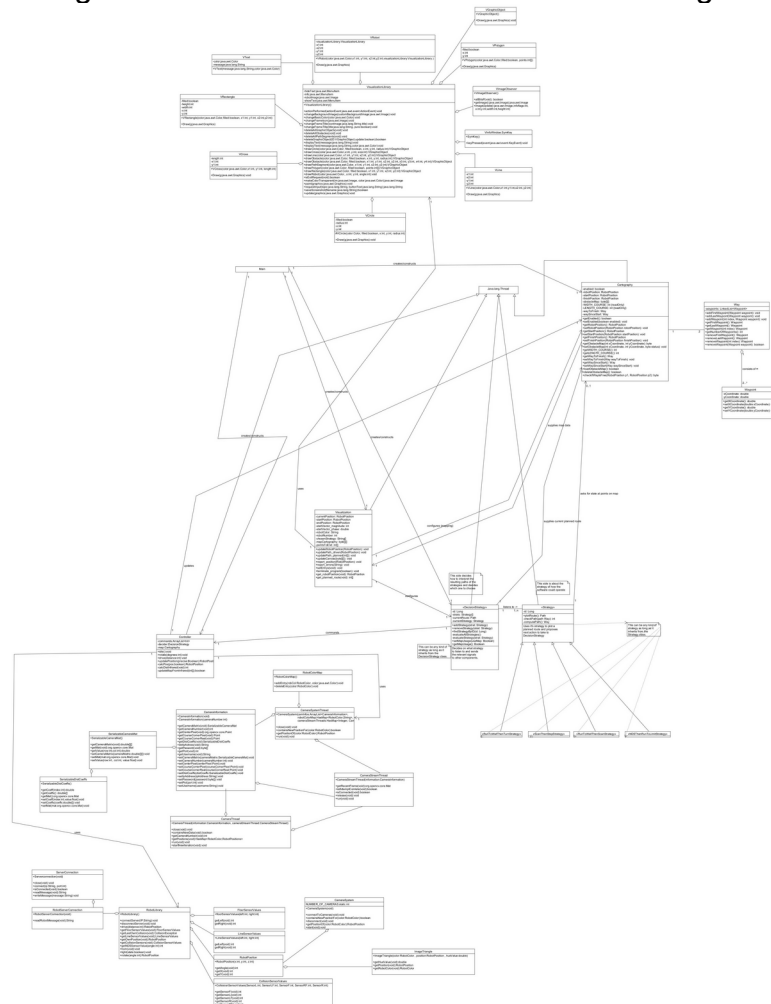
Entsprechen den im Pflichtenheft definierten Umgebungs- und Randbedingungen.

2 Grundlegende Entwurfsentscheidungen

Aufteilung der Module wurde an die vom IAS in den Aufgabenstellungen der vergangenen Blättern angegebenen Teilbereiche angelehnt, entsprechend die Unterteilung in die Module Steuerung, Strategie, Visualisierung und Kartographie.

3 Diagramm der Software-Systemarchitektur

Die grafische Darstellung der Komponenten in Form eines Klassendiagramms ist als separates PDF im Anhang zu diesem Dokument zu finden. Hier eine grobe Übersicht:



4 Softwarekomponenten

Steuerung: Umfasst die Steuerung des Roboters und damit die Aktoren und Sensoren dessen. Setzt die Befehle der Strategie um und sendet relevante Daten an die anderen

Module wie etwa der Kartographie (die Hindernisse die über Sensoren erkannt wurden) oder der Strategie (Roboter kurz vor Zusammenstoß).

Strategie: Erarbeitet durch Analyse aller ihr verfügbarer Daten die nächsten zu tätigen Schritte. Nach Festlegung dieser, steuert sie die anderen Module an um die erarbeiteten Schritte zu verfolgen, beispielsweise eine Rotation in eine Richtung um ein breiteres Bild der Sachlage zu erhalten, ein Update der genauen Position oder ähnliches.

Visualisierung: Zuständig für das Eingabefenster in dem, zu Beginn eines jeden Laufs, die für den Start relevanten Daten einzutragen sind. Sendet, nachdem die Daten eingetragen wurden, diese Daten an die relevanten Schnittstellen. Im weiteren Betrieb zuständig für die Darstellung der aktuellen Daten auf dem Bildschirm im entsprechenden Fenster.

Kartographie: Zuständig für die Erstellung einer Karte anhand der Daten die durch die Sensorik erfasst werden. Stellt diese Karte der Visualisierung und Strategie zur Verfügung um diese darzustellen oder zur Berechnung des geplanten Pfads zu nutzen.

5 Schnittstellendefinitionen

5.1 Schnittstelle Controller

Der Controller bildet die Hauptschnittstelle zu den vom IAS bereitgestellten Klassen und deren Funktionen, da diese für die Umsetzung von Steuerbefehlen zuständig ist.

5.2 Schnittstelle Visualisierung

Als Schnittstelle zum Nutzer werden hier, entsprechend den Anforderungen, keine Interaktionsmöglichkeiten während dem Lauf angeboten außer dem Schließen der Applikation.