

# Von C# nach Python: Software-Konzeptionierung einer robotergestützten Lagerverwaltung

Analyse bestehender Software und Konzeptionierung einer integrierten  
Python-Anwendung mit kameragestützten Validierungsprozessen in der  
Industrie 4.0-Plattform Modellfabrik  $\mu$ Plant

Lennart Schink

May 30, 2023

## Contents

<b>1</b>	<b>Einleitung und Motivation</b>	<b>2</b>
<b>2</b>	<b>Softwarearchitektur der bestehenden Software</b>	<b>3</b>
2.1	Lagerverwaltung 3.0 . . . . .	3
2.2	RFID Server . . . . .	3
2.3	Controller . . . . .	3
<b>3</b>	<b>Konzeptionierung einer integrierten Python Anwendung</b>	<b>4</b>
3.1	Konzepte zur Datenmodellierung . . . . .	4
3.2	Konzepte für Controller- und Serviceklassen . . . . .	4
3.3	GUI - Konzeptionierung . . . . .	4
3.4	Teilautomatisierte Code Dokumentation . . . . .	4
<b>4</b>	<b>Analyse zur Fehlerbehandlung</b>	<b>5</b>
<b>5</b>	<b>Ideensammlung zu kameragestützten Validierungsprozessen in der Lagerverwaltung</b>	<b>6</b>
5.1	Konzepte . . . . .	6
5.2	Abgeleitete Anforderungen an die Kamera . . . . .	6
5.3	Kameraauswahl . . . . .	6

# 1 Einleitung und Motivation

Das Institut für Mess- und Regelungstechnik an der Universität Kassel hat in den letzten Jahren eine Modellfabrik  $\mu$  Plant gebaut. Aus über 70 Einzelarbeiten ist ein modernes Industrie-4.0 Konzept geschaffen worden. Teil der  $\mu$  Plant ist ein Vollautomatisiertes Lager. Das Lager besteht aus einem abgetrennten Raum, dessen Zugang über eine Tür mit einem Türschalter überwacht ist. In diesen Bereich können Turtlebots einfahren. In dem abgetrennten Bereich steht ein Industrieroboter und ein Lagerregal mit ausgewiesenen 18 Lagerplätzen. Außerdem befindet sich neben einer Andockstation für den Turtlebot auch noch eine Werkbank.

Ein pneumatischer Greifer des Industrieroboters kann Paletten, die je mit bis zu zwei Bechern bestückt werden können, zwischen dem mobilen Roboter und dem Lagerregal frei bewegen. Von einem PC-Arbeitsplatz aus können mittels Software die Lagerprozesse überwacht werden. Außerdem kann im Fehlerfall eingeschritten werden und es können manuell Prozesse ausgelöst werden.

Die Software ist derzeit in 3 Teile aufgeteilt: Einerseits gibt es die Lagerverwaltung - die Hauptsoftware. Sie bildet die automatisierten Prozesse ab und verfügt über ein GUI welches das Lager mit seinem Bestand visualisiert. Daneben gibt es den Warehouse Controller, der dazu verwendet wird manuell Lagerprozesse auszulösen, und ein RFID-Tool was für manuelle RFID - Prozesse benutzt wird.

Mit dem Wechsel des Betriebssystems von Windows 7 auf Windows 10 ist die Kompatibilität der in C# implementierten Software nicht mehr gegeben. Außerdem laufen Teilfunktionen des Programms nicht fehlerfrei oder tolerieren keine Fehlbedienungen. Die Dreiteilung der Software ist im Allgemeinen auch nicht mehr erwünscht.

Diese Seminararbeit beschäftigt sich mit der Analyse der bestehenden Software: Es wird ermittelt, aus welchen Programmteilen und Funktionen die Software besteht. Aus den Erkenntnissen wird ein Konzept entwickelt wie die Drei Software Teile zusammengeführt werden könnten um so die Grundlage für eine Migration der Software nach Python zu schaffen.

Erkenntnisse aus der studentische Arbeit von [Hügler] sollen überprüft und vertieft werden um Anforderungen an Kameras und arUco Marker zu ermitteln, die später eine automatisierte Inventur ermöglichen sollen.

## **2 Softwarearchitektur der bestehenden Software**

### **2.1 Lagerverwaltung 3.0**

### **2.2 RFID Server**

### **2.3 Controller**

### **3 Konzeptionierung einer integrierten Python Anwendung**

#### **3.1 Konzepte zur Datenmodellierung**

#### **3.2 Konzepte für Controller- und Serviceklassen**

#### **3.3 GUI - Konzeptionierung**

#### **3.4 Teilautomatisierte Code Dokumentation**

## **4 Analyse zur Fehlerbehandlung**

## **5 Ideensammlung zu kameragestützten Validierungsprozessen in der Lagerverwaltung**

### **5.1 Konzepte**

### **5.2 Abgeleitete Anforderungen an die Kamera**

### **5.3 Kameraauswahl**