



Diplomand
Dozent
Projektpartner
Experte
Themengebiet

Janis Perren
Dr. Adrian Koller
Institut IME, CC Mechanische Systeme
Dipl. Ing. Paul Joachim Schüngel
Produktentwicklung & Mechatronik

Neuentwicklung des Postprozessors von RoboDK

Ausgangslage

Roboter sind aus der Industrie nicht mehr wegzudenken und erlangen zunehmend an Aufmerksamkeit. In den meisten Anwendungen befindet sich ein Entwickler neben einem Roboter, um diesen zu programmieren. Dies führt zu langen Standzeiten und erfordert physische Anwesenheit. Diese Art der Programmierung wird Online-Programmierung genannt. Die Offline-Programmierung wird mit einer vorherigen Simulation realisiert, in welcher die gesamte Applikation programmiert werden kann und diese nur noch auf den realen Roboter geladen werden muss, sobald sie fertiggestellt ist. RoboDK ist ein Programm, welches solche Simulationen von Robotern verschiedensten Herstellern ermöglicht. An der HSLU wird das Arbeiten mit RoboDK bereits gelehrt, allerdings gibt es Lücken in der Funktionalität der Schnittstelle zwischen dem Simulationsprogramm und dem Roboterprogramm. Die Schnittstelle ist dafür zuständig, aus dem Simulationsprogramm ein lauffähiges Roboterprogramm zu generieren. Diese Schnittstelle gibt es in diesem Projekt zu analysieren und mit einem geeigneten Konzept zu verbessern.

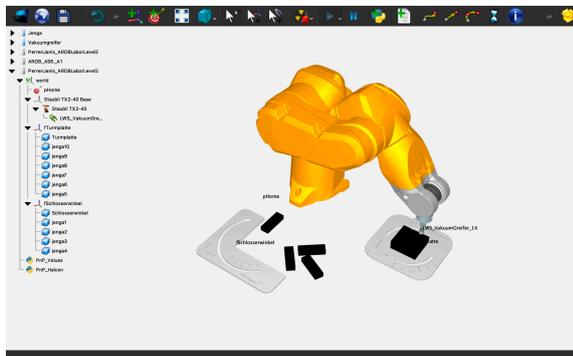


Abb. 1: Beispiel einer Simulation in RoboDK

Vorgehen

In einem ersten Schritt wurde recherchiert, wie die Schnittstelle genau arbeitet und inwieweit sie in der Theorie anpassbar ist. In einem zweiten Schritt wurde die Funktionalität der Schnittstelle getestet. Dazu wurden verschiedenste Simulationsprogramme als Use-Cases definiert und mit der Schnittstelle in Roboterprogramme umgewandelt. Damit konnte aufgezeigt werden, welche Funktionen richtig übersetzt werden, und welche nicht. Nach der Analyse konnten einige Konzepte erarbeitet werden, um die Schnittstelle zu verbessern oder gar zu ersetzen.

Ergebnis

Die Recherche über die Arbeitsweise der Schnittstelle hat ein Diagramm ergeben, welches in Abbildung 2 zu sehen ist. Die Schnittstelle wandelt das



Abb. 2: Flussdiagramm einer Generierung eines Roboterprogrammes in RoboDK

Simulationsprogramm zuerst in ein Zwischenprogramm um, und bezieht dann einen Postprozessor ein. Das Zwischenprogramm ist hierbei noch unabhängig vom gewählten Robotermodell und -hersteller. Der Postprozessor hat dann die Aufgabe, dieses unabhängige Zwischenprogramm in ein gewünschtes Format für den gewählten Roboterhersteller umzuwandeln.

Der erste Übersetzungsschritt (in Abb. 2 blau dargestellt) ist nicht manipulierbar. Das Simulationsprogramm wird in diesem Schritt linearisiert. Man stelle sich vor, jemand lässt das Simulationsprogramm laufen und notiert sich dabei jede Aktion, welche der Roboter in der Simulation ausführt. Diese Aktionen werden dann im Zwischenprogramm aufgelistet. Allerdings ist dort nichts mehr von der eigentlichen Programmierung ersichtlich. Es ist daher nicht möglich, die Logik des Simulationsprogrammes im Roboterprogramm zu rekonstruieren. Ausserdem befinden sich im Zwischenprogramm nur explizit die Aktionen, welche von RoboDK übersetzt werden sollen. Möchte man also eine für einen Roboterhersteller spezifische Aktion ausführen, ist das nicht möglich. Es zeigten sich auch mehrere Schwachstellen im Schritt zwei (in Abb. 2 grün dargestellt), welcher den Postprozessor betrifft. Einige grundlegende Funktionen, welche für ein Roboterprogramm nötig sind, wurden nicht richtig umgewandelt, oder gar ignoriert.

Zur Lösung dieser Probleme wurde ein Konzept entwickelt, welches den Postprozessor optimiert und eine Python-Library zur Verfügung stellt, welche im Simulationsprogramm genutzt werden kann. Die Python-Library hat die Aufgabe, einerseits den Programmierer bei der Entwicklung durch Funktionen zu unterstützen und hat andererseits die Aufgabe, bestimmte Befehle durch eine Manipulation von Schritt 1 an den Postprozessor zu übergeben. Damit ist es nun möglich, spezielle Aktionen, welche RoboDK eigentlich nicht übersetzt, in das Roboterprogramm schreiben zu lassen.