



Fachpraktikum Gruppe 3

Erstellung von Steuersoftware zur Navigierung eines Roboters durch einen Hinderniskurs unter simultaner Kartografierung des selbigen

Waldemar Repp, Michael Streib, Tim Braun, Marcel Sauter

Pflichtenheft

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



Universität Stuttgart
Institut für Automatisierungstechnik
und Softwaresysteme

Dokument Versionsverwaltung

Vers ion	Autor	QS	Datum	Status	Änderungen
0.1	WaRe	MaSa, MiSt,TiBr	21.11.20	Durch QS.	Erstellung
0.2	WaRe	MaSa, MiSt,TiBr	21.11.20	Durch QS.	Vervollständigung Qualitäts- Zielbestimmung.
0.3	WaRe	MaSa, MiSt,TiBr, NaHa	23.11.20	Durch QS.	Korrekturen aus QS einbinden
0.4	WaRe	MaSa, MiSt,TiBr, NaHa	24.11.20	Eingereit.	Kleinere Korrekturen

0 Inhaltsverzeichnis

0	INHALTSVERZEICHNIS	3
1	ZIELBESTIMMUNG	4
1.1	Musskriterien	4
1.2	Wunschkriterien	4
1.3	Abgrenzungskriterien	4
2	EINSATZ	4
2.1	Anwendungsbereiche	4
2.2	Zielgruppen	4
2.3	Betriebsbedingungen	4
3	UMGEBUNG	4
3.1	Software	4
3.2	Hardware	4
3.3	System-Schnittstellen	4
4	FUNKTIONALE ANFORDERUNGEN	4
5	NICHTFUNKTIONALE ANFORDERUNGEN	4
6	ANFORDERUNGEN AN DIE BENUTZUNGSSCHNITTSTELLE	5
7	QUALITÄTS-ZIELBESTIMMUNG	5
8	 Globale Testszenarien/-fälle	6
9	ENTWICKLUNGSUMGEBUNG	7
9.1	Software	7
9.2	Hardware	7
9.3	Orgware	7
9.4	Entwicklungs-Schnittstellen	7
10	DURCHFÜHRUNG	7
11	ERGÄNZUNGEN	7

1 Zielbestimmung

1.1 Musskriterien

Den Kundenwünschen wurden folgende Musskriterien entnommen:

- Der Roboter muss Hindernisse erkennen und an diesen vorbei manövrieren
- Der Roboter vermeidet Wände, Hindernisse und die Grenzen des Parkours
- Auf dem Bildschirm des Steuerrechners müssen die Grenzen des Kurses, die Position des eigenen Roboters, der Ausgangspunkt, das Ziel sowie der geplante Weg visualisiert werden.
- Der Roboter darf keine Eingaben nach Erhalt des Startsignales erhalten
- Nur beim Start dürfen dem Roboter Daten in Form von Parametern, einer Datei oder über eine grafische Oberfläche gegeben werden und diese sind auf den Startpunkt, den Endpunkt, einen Startvektor, die zu nutzende Strategie, die Roboterfarbe und seine Nummer beschränkt.
- Für den Fall das der Roboter mehr als ein mal eine Hinderniskonstellation zu durchfahren hat, muss der Roboter diese kartografieren und in einer Datei speichern. Die Nutzung dieser Karte muss nach dem ersten Rennen für weitere Fahrten nutzbar sein. Die Verwendung der Karte muss optional sein.
- Es darf nicht möglich sein, dass die Software durch falsche Benutzereingaben abstürzt.
- Die Benutzung der Software muss einfach sein und durch ein kurzes Benutzerhandbuch erlernbar sein.
- Es ist der GIT-Server des TIK für das Konfigurationsmanagement zu benutzen.
- Die Verwendung der Sensoren außer der Infrarotabstandssensoren muss optional sein. Der NDS ist ein Beispiel für optionale Sensorik.
- Wird die Software beendet oder abgebrochen müssen alle dazugehörigen Javaprozesse beendet werden.

1.2 Wunschkriterien

Diese Kriterien wurden vom Kunden als erwünscht beschrieben:

- Die Software sollte die vorhandenen Bibliotheken für Visualisierung und Steuerung nutzen

Diese Kriterien sind vom Team als optional und förderlich beschrieben worden:

- Die Nutzung des NDS-Sensors des Roboters kann großflächige Erfassung des Kurses und somit effiziente Wegfindung ermöglichen.
- Optimierte Ansteuerung des Roboters könnte weiteren Zeitgewinn erlauben in dem nicht zwischen Vorwärtsfahrt und Drehung unterschieden wird und stattdessen Kurvenfahrt implementiert werden könnte.

1.3 Abgrenzungskriterien

Diese Kriterien definieren welche Fälle zu beachten sind und welche nicht:

- Hindernisse dürfen aneinander gestellt werden. Stehen Hindernisse frei, also nicht aneinander, so haben diese immer einen Abstand der mindestens der Länge eines schmalen Hindernisses entspricht.

2 Einsatz

2.1 Anwendungsbereiche

2.1.1 Hindernissparkour

- Die Software ist nur für den Hindernissparkour in der vom Kunden spezifizierten Version (siehe Musskriterien) entwickelt.
Eine andere Nutzung wird bei der Entwicklung nicht berücksichtigt.

2.2 Zielgruppen

2.2.1 Mitarbeiter des IAS

- Die Software wird für den Kunden (das Institut für Automatisierungstechnik der Universität Stuttgart) unter Berücksichtigung der festgelegten Anforderungen entwickelt.
- Eine Nutzung durch andere Personen, etwa Personen die nicht das nötige technische Verständnis besitzen, wird nicht unterstützt. (siehe Musskriterien)

2.3 Betriebsbedingungen

2.3.1 Hardware voll funktionell

- Die korrekte Funktionsweise der Software kann nur sichergestellt werden sofern das Hardwaresystem korrekt funktioniert.
- Dies bedeutet dass die Recheneinheiten sowie Sensorik und Aktoren, wie unter anderem die Antriebsmotoren und Abstandssensoren, ihren Dokumentationen entsprechend agieren.
- Aufgrund des Wunsches des Kunden wird die Software derart entwickelt dass der Roboter seine Aufgabe auch dann erfüllt wenn dieser lediglich seine Infrarotabstandssensoren als Sensoren nutzen darf.

3 Umgebung

3.1 Software

- Die Entwicklung der Software zur Steuerung des Roboters wird unter Nutzung der vom IAS zur Verfügung gestellten Bibliotheken durchgeführt.
- Dies bedeutet unter anderem auch, dass der Quellcode der Software in Java (JAVA SE) erstellt wird.
- Die zur Erzeugung des Programmcodes genutzten Werkzeuge sind nicht vom Kunden vorgeschrieben. Die Wahl dieser Werkzeuge liegt im Ermessen des Entwicklungsteams (IAS FaPra Team 3).
- Die Softwareschnittstellen sowie das Verhalten der Software mit der die zu erstellende Robotersteuerungssoftware interagiert, unter anderem das Kamerasystem sowie die Software, die für die Übertragung der von der Steuerungssoftware erzeugten Befehle zuständig ist, werden als unveränderlich festgelegt.

Sofern Änderungen an dieser Software durchgeführt werden müssen, wird seitens des IAS sichergestellt, dass diese Systeme sich weiterhin dermaßen verhalten wie es in den Dokumentationen, die zu Beginn des Projektes verteilt wurden, beschrieben wird.

- Sofern durch Änderungen der Software durch das IAS Verzögerungen beim Entwicklungsbetrieb oder Testbetrieb bzw. eine Minderung der möglichen Gelegenheiten für Tests am Roboter entstehen, kann es eventuell zu dem Fall kommen dass das Projektteam die Software nicht rechtzeitig zu voller Reife und damit ausreichendem Funktionsumfang bringen kann. Sofern dieses Szenario aufzutreten droht, setzt sich das Team mit dem IAS als Auftragsgeber in Verbindung.

3.2 Hardware

- Die Hardware des Roboters sowie anderer, peripherer Hardware die nötig ist zum korrekten Betrieb des Roboters in seiner zu erfüllenden Funktion, wird als unveränderlich festgelegt. Sofern das IAS Änderungen, beispielsweise Austausch von defekten Teilen, an diesen Hardwarekomponenten durchzuführen hat, muss von Seiten des IAS sichergestellt werden, dass die korrekte, damit also nach Dokumentation festgelegten Parametern und Abläufen, Funktion der entsprechenden Komponenten sichergestellt wird.
- Die Hardware des Roboters sowie der peripheren Systeme hat in den für diese Komponenten spezifizierten Umgebungsbedingungen zu erfolgen. Unzulässige Luftfeuchtigkeit, Temperaturen oder Temperaturschwankungen die außerhalb der technischen Spezifikationen der Hardwaresysteme und ihrer Teilkomponenten liegen, sind ungültig. Die Bereitstellung einer Umgebung nach diesen Spezifikationen sind vom IAS bereitzustellen.

3.3 System-Schnittstellen

Der Roboter hat, unter Berücksichtigung seiner Aufgaben und der Anforderungen, folgende Systemschnittstellen mit denen dieser zu interagieren hat:

- Steuercomputer zu Roboter Schnittstelle
- Steuercomputer zu Kamerasystem Schnittstelle
- Mensch-Computer-Schnittstelle

Für alle Schnittstellen wird, wie oben bereits beschrieben, festgelegt dass diese als nicht-veränderbar angesehen werden. Diese Schnittstellen wurden vom IAS bereits festgelegt und die Interaktion mit diesen wurde durch vom IAS bereitgestellten Bibliotheken implementiert bereits implementiert. Entsprechend wird vorausgesetzt dass diese Bibliotheken Ihre Funktionen korrekt erfüllen und sind damit nicht Bestandteil der zu entwickelnden Software. Sie, die Schnittstellen und Bibliotheken, sind lediglich zu nutzen.

4 Funktionale Anforderungen

Steuerung

- /PFA10/ Der Roboter hat, nachdem dieser das Startsignal erhalten hat, einen Weg zum Ziel zu finden. (Lastenheft A7 und A1)
- /PFA11/ Nach dem Startsignal sind keine interaktiven Eingaben erlaubt. (Lastenheft A7)
- /PFA12/ Die Eingaben seitens des Nutzers sind auf Eingabeparameter bei Softwarestart beschränkt. (Lastenheft A8)
- /PFA13/ Diese Eingaben können durch Eingabe über ein grafisches Interface am Steuerrechner, durch Auslesen einer Datei oder Parameterübergabe bei Programmaufruf erfasst werden. (Lastenheft A8)
- /PFA14/ Die Informationen dieser Eingaben sind auf den Startpunkt und den Endpunkt als Koordinaten, einen Startvektor, die zu nutzende Strategie, die Farbe des Roboters sowie dessen Nummer beschränkt. (Lastenheft A8)

Strategie

- /PFA20/ Die Software hat einen Weg für den Roboter zu planen. Hierbei müssen Anforderungen erfüllt werden. (Lastenheft A4)
- /PFA21/ Der zu planende Weg muss die vorgegeben Grenzen berücksichtigen. (Lastenheft A2)
- /PFA22/ Der zu planende Weg muss um Hindernisse und Wände herum führen. (Lastenheft A4, A3)
- /PFA23/ Der Roboter muss auch dann einen Weg zum Ziel finden können falls dieser lediglich seine Infrarotabstandssensoren nutzen darf.

Kartographie

- /PFA30/ Die Software muss während einem Durchlauf eine Karte des Parkours mit den erkannten Hindernissen erstellen und abspeichern können. (Lastenheft A11)
- /PFA31/ Diese Karte muss nach dem ersten Durchlauf für die kommenden Fahrten nutzbar sein. (Lastenheft A11)
- /PFA32/ Die Nutzung der Karte muss abschaltbar sein. (Lastenheft A11)

Sonstiges

- /PFA40/ Wird die Software beendet müssen alle dazugehörigen Javaprozesse beendet werden.
- /PFA41/ Die Software darf bei falscher Bedienung durch den Nutzer nicht abstürzen.

5 Nichtfunktionale Anforderungen

- /PNA10/ Es ist der dem Team zugewiesene GIT-Server des TIK zu nutzen.
 /PNA20/ Es sind die vorhandenen Bibliotheken für Visualisierung und Steuerung zu nutzen.

6 Anforderungen an die Benutzungsschnittstelle

Visualisierung

- /PBA10/ Die Steuersoftware hat ein grafisches Interface auf dem Steuercomputer anzuzeigen das folgende vorgegeben Daten anzeigt und gewisse Interaktionen bereitstellt. (Lastenheft A6)
 /PBA20/ Das Fenster hat die Grenzen des Parkours, die Position des Roboters, den Ausgangspunkt sowie den von der Software geplanten Weg darzustellen. (Lastenheft A6)
 /PBA30/ Die Software darf bei falscher Bedienung durch den Nutzer nicht abstürzen.
 /PBA31/ Die Software muss, nachdem das kurze Benutzerhandbuch gelesen wurde, bedienbar sein.
 /PBA40/ Das Programm soll durch Klicken des X Symbols des Fensterrahmens beendbar sein.

7 Qualitäts-Zielbestimmung

	sehr hoch	hoch	normal	nicht relevant
Funktionalität	x			
Richtigkeit	x			
Sicherheit (Security)				x
Interoperabilität		x		
Zuverlässigkeit	x			
Reife		x		
Fehlertoleranz			x	
Wiederherstellbarkeit			x	
Sicherheit (Safety)			x	
Benutzbarkeit	x			
Verständlichkeit			x	
Erlernbarkeit			x	
Bedienbarkeit	x			
Effizienz			x	
Zeitverhalten			x	
Verbrauchsverhalten			x	

Änderbarkeit	x
Analysierbarkeit	x
Modifizierbarkeit	x

Kommentare bezüglich der Qualitäts-Zielbestimmung:

- Aufgrund der zugrunde liegenden Anforderungen sind die Faktoren Funktionalität, Richtigkeit, Zuverlässigkeit sowie Benutzbarkeit, Bedienbarkeit und Analysierbarkeit zunächst als sehr hoch eingestuft. Explizit genannt seien die Anforderungen zur Bedienbarkeit und dem gewünschten Verhalten des Systems.
- Die hohe Bewertung der Analysierbarkeit folgt aus der Zusammenarbeit mehrerer Personen an einer Software sowie der Interaktion dieser Software mit anderen Systemen sowie der nur seltenen Möglichkeit von realen Tests aufgrund der Natur des Systems (Hardwaresystem ohne Simulation und Begrenzung auf Teststandort Labor)
- Da der Roboter die Teststrecke alleine abfahren wird und während dessen keine anderen Teams ihre Software ausführen können ist security ein zu vernachlässigender Aspekt. Verschlüsselungen und Authentifizierungen sind entsprechend nicht nötig.

8 Globale Testszenarien/-fälle

- Während Entwicklung gelegentliche Tests im Labor durch Qualitätssicherung.
- Abnahme durch Test im Labor am Ende der Entwicklungszeit durch Mitarbeiter des IAS.

9 Entwicklungsumgebung

9.1 Software

- Von Teammitgliedern individuell gewählte Editoren für Entwicklung von Programmen unter Nutzung von Java SE
- Bibliotheken vom IAS bereitgestellt oder Standardbibliotheken der Java SE
- Java virtuelle Maschine zur plattformunabhängigen Ausführung des Codes

9.2 Hardware

- Tests auf bereitgestellter Hardware des IAS im Labor (unter anderem Roboter, Kamerasystem, Steuerrechner, Netzwerk und andere)
- Softwareentwicklung auf privaten Rechnern der Studenten

9.3 Orgware

- Teaminterne Kommunikation über Slack (www.Slack.com) und Skype (www.Skype.com)
- Ticketsystem Jira (www.Atlassian.com/software/jira)
- Kommunikation mit IAS-Mitarbeitern über ILIAS

9.4 Entwicklungs-Schnittstellen

- Vom IAS bereitgestellte Bibliotheken und Java SE Standardbibliotheken

10 Durchführung

Die Arbeit wird nach dem IAS-Vorgehensmodell (Modell zur Softwareentwicklung) durchgeführt werden.

Der Stand der Arbeit und die Ergebnisse werden in regelmäßigen Abständen (ca. alle 2 Wochen) mit den Betreuern diskutiert.

Bei der Durchführung der Arbeit und der Anfertigung der Ausarbeitung werden die Richtlinien des IAS beachtet.

11 Ergänzungen

Keine.