

Praktikum Software-Engineering 1

Aufgabenblatt 1

Prof. Dr. Thomas Lehmann, <thomas.lehmann@haw-hamburg.de>

Lerning Outcomes

Sie ...

- ... identifizieren den Systemkontext von einem System.
- ... entwerfen und testen eine Wrapper-Klasse.
- ... entwerfen automatisierte Tests auf Basis von Äquivalenzklassen und Grenzwerten.
- ... stellen geeignete Testumgebungen her.

Bearbeitungshinweise

- Die Bearbeitung der Aufgaben findet in **3er-4er Gruppen** statt.
- Die bearbeiteten Lösungen werden während der Praktikumsstunde diskutiert und abgenommen. Dazu werden Sie abwechselnd Ihre Lösungen vorstellen und dazu befragt werden.
- Es gibt **100% Anwesenheitspflicht** beim Praktikum. Beim Fehlen wegen z. B. Krankheit müssen Atteste eingereicht werden und Nachholtermin werden dann vergeben.
- Fügen Sie textuelle Ergebnisse in das Template ein. Code ist jeweils in einem Zip-Paket beizufügen. Alles zusammen muss innerhalb von einer Woche nach dem Praktikumstermin als ein Zip-Paket hochgeladen werden.

Teil 1: Systemkontext

Für ein System soll auf Basis der Fallbeschreibung ein Systemkontext identifiziert und dokumentiert werden.

Fallbeschreibung Autonomes Fahren

Studierende der HAW Hamburg wollen am studentischen Wettbewerb zum autonomen Fahren teilnehmen. In dem Wettbewerb werden Modellautos im Maßstab 1:10 eingesetzt und es sind verschiedenen Aufgaben zu erfüllen. Weiterhin müssen einem Team von Fachleuten aus der Industrie die eingesetzten Technologien vorgestellt werden. Die Aufgabenstellung und erste Ideen für das Systemdesign sehen wie folgt aus:

Das Fahrzeug muss autonom einer Fahrbahn mit zwei Fahrspuren folgen. Die Fahrbahn wird durch weiße Linien auf einem schwarzen Untergrund begrenzt. Als Form kann eine liegende Acht angenommen werden. Im Kreuzungsbereich sind Stopp-Balken aufgezeichnet. Es wird kurz angehalten. Es soll in seiner Spur bleiben, es sei denn es trifft auf feststehende Hindernisse. Ziel ist es den Kurs mit möglichst hoher Geschwindigkeit bei möglichst vielen Runden zu absolvieren. Die Fahrt ist beendet, wenn eine Fahrbahnbegrenzung zum zweiten Mal überfahren wurde oder an der Haltelinie nicht gehalten wurde. Start des Fahrzeuges erfolgt durch einen der Schiedsrichter. Vor dem Start dürfen am Fahrzeug keine weiteren Modifikationen vorgenommen werden.

In einem zweiten unabhängigen Wettbewerbsteil soll autonom rückwärts eingeparkt werden. Das Fahrzeug fährt eine Gerade entlang. Findet es eine passende Parklücke, so soll rückwärts in die Lücke eingeparkt werden.

Relevante Systemdaten, wie zum Beispiel die Odometrie, werden lokal erfasst und zusätzlich auf den Entwicklungsrechner übertragen. Dort werden sie gespeichert und in einem Cockpit dargestellt. Die Systemdaten können auch über einen Web-Server auf dem Fahrzeug abgerufen werden. Die Betriebsbereitschaft wird angezeigt.

Hindernisse als auch „parkende Autos“ sind weiße Kisten mit ca. 15 cm Kantenlänge.

Im Notfall muss mit einer Fernsteuerung eingegriffen werden.

Hinweis: Das Entwicklungsprojekt im Rahmen des Praktikums bezieht sich nur auf das Fahrzeug. Entsprechend fehlen hier einige implizite Requirements. Ergänzen Sie diese ggf. sinnvoll!

Vorbereitung

Erstellen Sie für das autonome Fahrzeug ein Systemkontextdiagramm. Ergeben sich mehrere Kontexte, so sind entsprechend mehrere Diagramme zu erstellen. Die Diagramme sollen zum Beginn des Praktikums in mit Beamer präsentierbarer Form vorliegen¹.

Im Praktikum

Die erarbeiteten Diagramme werden im Plenum diskutiert.

¹ Es darf handschriftlich gearbeitet werden, das Diagramm sollte aber lesbar, eingescannt vorliegen.

Nacharbeiten

Im Rahmen der Nacharbeiten sind das bzw. die Diagramme zu überarbeiten und in den Praktikumsbericht einzufügen.

Teil 2: Boundary-Classes und Tests

Für das System müssen auf Werte aus der Umgebung geeignet zugegriffen werden. Hierzu werden Boundary-Classes (Wrapper) und Service-Classes eingesetzt. In dieser Aufgabe wird exemplarisch eine Boundary-Class auf Basis eines Fallbeispiels entworfen und eine Service-Class getestet.

Fallbeispiel Sensordaten

Zur Abstandsmessung werden im Fahrzeug Infrarotabstandssensoren verbaut, die mittels Triangulation die Entfernung bestimmen. Aus den Entfernungen soll dann nur der Abstand als close, middle, far, off bestimmt werden.

Der Sensor kann Entfernungen bis maximal 30 cm bestimmen. Im Bereich unterhalb von 4 cm arbeitet dieser nicht mehr zuverlässig. Der Sensor liefert diskrete Werte d im Bereich von 0-255. Die Werte werden nach folgender Formel bezogen auf den Abstand x [cm] berechnet:

$$d = \left\lfloor \left(1 - \cos\left(\frac{x}{30} * \frac{\pi}{2}\right)\right) * 265 - 5 \right\rfloor$$

Werte $d < 0$ werden zu 0.

Die Abstandssensoren werden an zwei verschiedenen Positionen im Fahrzeug verbaut. Die Service-Klasse muss das berücksichtigen, indem ihr über den Mode die Position übergeben wird.

- Mode A: Im Abstand von 8 cm bis 18 cm wird der Abstand als middle angesehen, darunter als close. Im Abstand bis 25 cm als far und darüber hinaus als off.
- Mode B: Bis einschließlich 5 cm als close, im Abstand ab 14 cm als off. Bis 10 cm als middle. Rest far.

Vorbereitung

Definieren Sie kurz in Textform die Aufgabe der Service-Klasse, die eine Entfernung in [cm] in die Entfernungsklassen entsprechend des Modes umrechnet. Erstellen Sie die Testspezifikation für die Service-Klasse. Die Testspezifikation wird zu Beginn des Praktikums kurz vorgestellt und diskutiert.

Im Praktikum

Testen Sie die im Praktikum gegebene Service-Klasse gegen Ihre Testspezifikation. Implementieren Sie dazu auf Basis Ihrer Testspezifikation die nötigen Unit-Tests.

Erstellen Sie einen Wrapper, der die diskreten Werte wieder in Abstände in [cm] umrechnet und eine geeignete API zur Verfügung stellt. Entwerfen Sie eine geeignete Testumgebung und testen Sie den Wrapper. Beachten Sie, dass das Protokoll der Schnittstelle auch mit getestet werden soll.

Deliverables des Praktikums

Die Dokumentation des Systemkontextes, die Aufgabenspezifikation der Service-Klasse, die Testspezifikationen, die Aufgabenspezifikation des Wrappers soll in einem Dokument zusammengefasst werden. Das Dokument wird mit dem erstellten Code als ein Paket abgegeben.