

Praktikum Software-Engineering

Aufgabenblätter 3 und 4

Prof. Dr. Zhen Ru Dai <zhenru.dai@haw-hamburg.de>
Raum 7.86b, Tel. 040/42875-8153

Bearbeitungshinweise

- Die Bearbeitung der Aufgaben findet in **festen Dreiergruppen** statt.
- Es gibt **100% Anwesenheitspflicht** beim Praktikum. Beim Fehlen aufgrund von Krankheit ist ein ärztliches Attest vorzulegen. Ein Nachholtermin wird dann vereinbart.

Wichtige Information zum Ablauf dieses Praktikumsblatts
Die Aufgaben und Dokumentation sind zuhause vollständig vorzubereiten. In dem Praktikum werden wir über Ihre Ausarbeitungen diskutieren.

Ziel dieses Praktikums

Im Rahmen zweier Praktikumsterminen sollen Sie Reverse-Engineering (von Code zu Modell) für eine vorgegebene SW durchführen. Nach einer genauen Analyse soll das System um weitere Features erweitert werden. Die Implementierung soll als anschließend getestet werden.

Problem

In einer Produktionsanlage werden gefertigte Elemente bereits auf Verunreinigungen untersucht und entsprechend aussortiert. Bei der Produktion kommt es vereinzelt zu Positionierungsfehlern (Vergleiche Abbildung 2 und 3). Durch die Prüfung der Höhenlinie sollen diese fehlerhaften Elemente aussortiert werden, indem der Weitertransport verweigert wird.

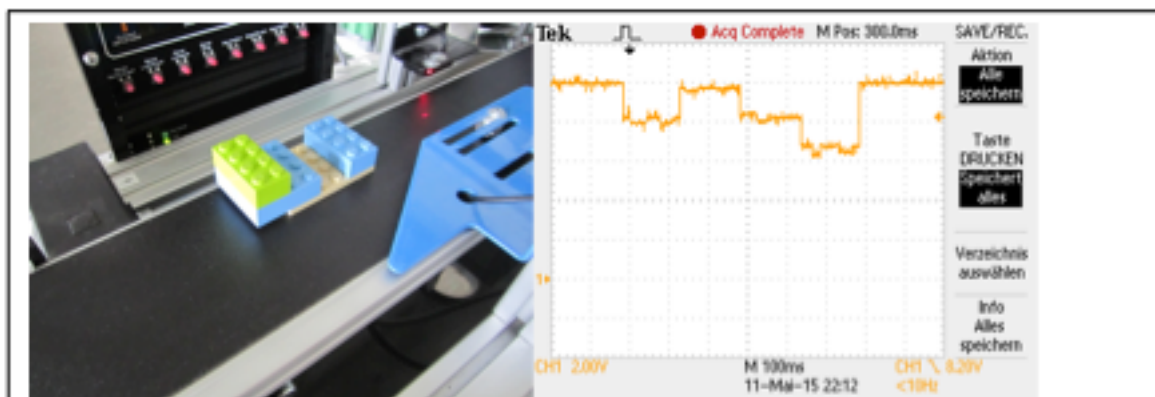


Abbildung 1: Korrekt gefertigtes Element und Signalverlauf des Höhengensors

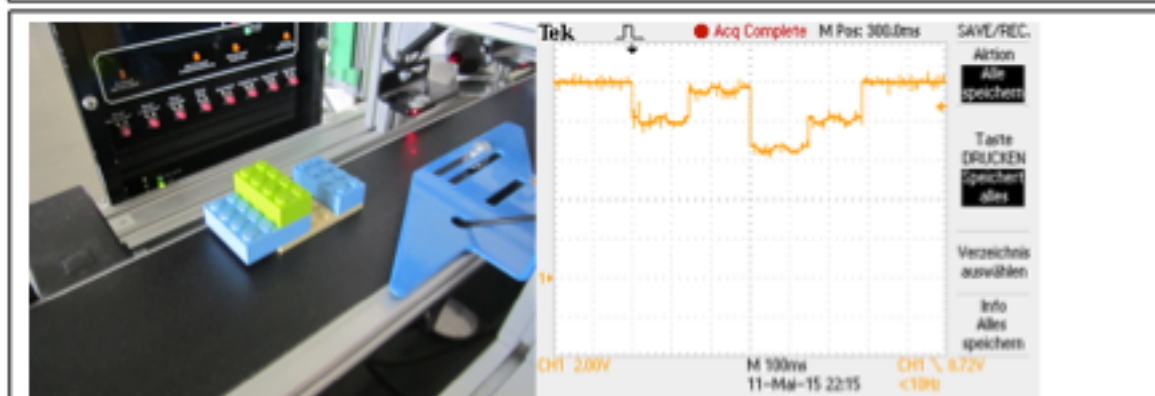


Abbildung 2: Versatz eines Bausteins sowie der Signalverlauf des Höhengensors

Die Steuerung der Aussortierung der Elemente erfolgt durch eine bestehende Steuerungssoftware für die Anlage. Es gibt einen Prototyp der Anlage, der einen Zugriff auf die Werte des Höhensensors erlaubt. An diesem Prototyp wurden in Testläufen mit den fehlerhaften Elementen die Werte des Höhensensors sowie der Lichtschranken aufgezeichnet.



Bei der Besprechung mit der Entwicklergruppe für die Steuerungssoftware wurden zwei Schnittstellen besprochen:

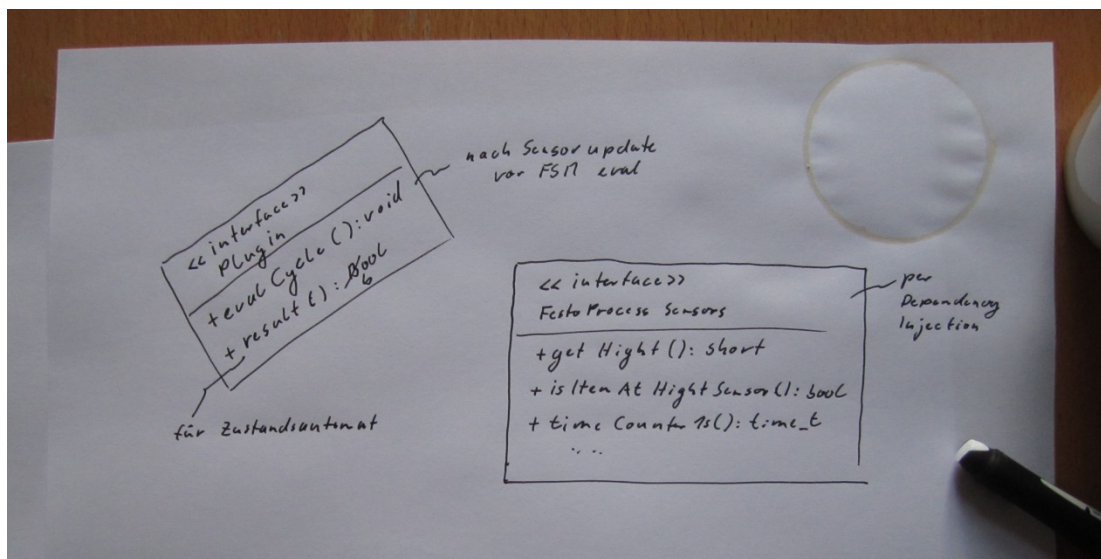


Abbildung 4: Abfotografierte Notizen der anderen Entwicklergruppe

Zum einen eine Schnittstelle für den Zugriff auf den Sensor, zum anderen die Schnittstelle für den Aufruf der von Ihnen zu entwickelnden Komponente.

Feature-Erweiterung:

Es ist eine Komponente/Klasse zu entwickeln, die fehlerhafte Elemente aufgrund der Sensordaten erkennt. Das Ergebnis der Erkennung soll durch den Steuerungsautomaten ausgewertet werden.

Das gewünschte zusätzliche Verhalten: Wird durch die Kontrolle der Höhenlinie ein fehlerhaftes Element identifiziert, so soll das Laufband das Element wieder zum Anfang zurück transportieren. Die rote Lampe der Ampel blinkt. Wird das Element entnommen, geht die Anlage wieder in den normalen Betrieb (Prüfung des nächsten Elementes) über.

Es wird auf der Konsole ausgegeben und in eine Log-Datei geschrieben, welche Art von Fehler das fehlerhafte Element aufweist.

Aufgabe ist es, die Komponente für die Fehlererkennung zu entwickeln sowie den bestehenden Automaten zu modifizieren. Die Komponente soll dabei unabhängig von der Anlage entwickelt und getestet werden.

3. Praktikumstermin (Reverse-Engineering und Feature Erweiterung):

Im Rahmen dieses Termins sollen Sie die mitgelieferte SW genau analysieren und erweitern (siehe Verzeichnis „*Sources_orig*“). Dabei führen Sie die Struktur und das Verhalten der SW in ihr Design-Modell zurück und führen die Feature-Erweiterung (s.o.) auf dem Design-Modell durch.

Vorgehensweise:

1. Reverse Engineering Struktur:

- Erstellen Sie anhand der gegebenen SW das Datenmodell (UML Klassendiagramm).
- Analysieren Sie das Datenmodell und leiten Sie aus dem Datenmodell die SW Architektur (SysML Block Diagram) ab.

Tipps:

- Unterscheiden Sie die Klassen in Interface Klassen und normale Klassen. Interface Klassen verbinden Architektur Blöcke. Stellen Sie fest, welche Klasse Interface Provider Klassen bzw. Consumer Klassen sind (Erinnern Sie sich an PR2). Klassen gehören zum gleichen Architektur Block, wenn keine Interface Klasse zwischen ihnen definiert ist.

- Diskutieren Sie über Vor- und Nachteile dieses vorgegebenen Designs. Denken Sie vor allem an die Design-Prinzipien „Separation of Concerns“ und „Information Hiding“.

2. Reverse Engineering Verhalten:

- Erstellen Sie anhand der gegebenen SW ein Verhaltensmodell mit Hilfe von UML State Machine.

3. Feature Erweiterung:

- Erweitern Sie die Architektur um das Zusatzfeature.
- Erweitern Sie das UML Klassendiagramm passend zu der erweiterten Architektur.
- Erweitern Sie die State Machine um das Zusatzfeature.

Abgaben: Erstellen Sie ein Design Dokument mit allen von Ihnen erstellten Diagrammen, und diskutieren Sie über das alte Design hinsichtlich den Design Kriterien *Separation of Concern*, *Erweiterbarkeit* und *Testbarkeit*. Erklären Sie Ihre Design-Erweiterung.
