

Praktikum Software Engineering 1

3. Reaktive Systeme

Learning Outcome des Praktikums

Die Studierenden können zu einem Fallbeispiel strukturiert reaktive Software entwerfen und implementieren, indem sie

- 1.) im Systemkontext das Ein-/Ausgabeverhalten analysieren,
- 2.) das Verhalten mittels Automaten modellieren,
- 3.) Testsequenzen ermitteln und modellieren,
- 4.) das modellierte Verhalten mittels eines Pattern für gegebene Schnittstellen implementieren und
- 5.) die Implementierung automatisiert mittels Unit-Tests verifizieren.

Ablauf des Praktikums

Lösen Sie als **Vorbereitung** die Teilaufgaben. Verteilen Sie dazu die Teilaufgaben im Team, bzw. reichen Sie die Ergebnisse der Teilaufgaben an das nächste Teammitglied weiter. Setzen Sie sich untereinander Deadlines für die Bearbeitung der Teilaufgaben!

Im Praktikumstermin stellt jedes Team kurz Ihre Lösung dem Plenum vor. Das Plenum führt ein informelles Review durch und offene Fragen können geklärt werden. Das Review bezieht sich ggf. nur auf Teile des Lösungskonzeptes. Der Rest des Praktikumstermins soll für Korrekturen und/oder Nacharbeiten verwendet werden.

Deliverables des Praktikums

Alle Diagramme sind zusammen mit kurzen Kommentaren als Bericht abzugeben. Wurden die Diagramme mit einem Tool erstellt, so sind die Dateien mit abzugeben und die Diagramme im Bericht zu referenzieren.

Die erstellten Sources sind mit abzugeben.

Fallbeschreibung

In einer Fertigungszelle soll die Feuchtigkeit reguliert werden und sich dadurch innerhalb der Grenzen `HumidityMin` und `HumidityMax` bewegen. Die Befeuchtung bzw. die Trocknung soll über eine State Machine gesteuert werden. Die State Machine soll die Trocknung bzw. Befeuchtung wie folgt steuern:

Nach dem Start wird die Feuchtigkeit überwacht. Unterschreitet die Feuchtigkeit den unteren Grenzwert, so wird das Signal `SprayOn` an die Befeuchtungsanlage geschickt. Erreicht die Feuchtigkeit wieder den Grenzwert, so wird die Befeuchtung durch Übermittlung des Signals `SprayOff` wieder abgeschaltet. Die Aktivierung der Befeuchtung soll zusätzlich über die Lampe A angezeigt werden.

Übersteigt die Feuchtigkeit den oberen Grenzwert so wird zunächst das Tor (Gate) geschlossen. Während sich das Tor schließt, ist die Lampe B eingeschaltet. Meldet das Tor zurück, dass es geschlossen ist, so werden zwei Pumpen (Zu- und Abluft), die wie das Tor über ein Netzwerk verbunden sind, eingeschaltet. Beide Pumpen melden jeweils zurück, dass sie sich erfolgreich aktiviert haben. Melden jeweils beide oder eine nicht innerhalb von 5 s nach Aktivierung den Erfolg, so soll in einen Error-Zustand übergegangen werden, beide Pumpen deaktiviert und das Tor geöffnet werden. Danach wartet das System auf eine manuelle Bestätigung und geht dann wieder in die Überwachung der Feuchtigkeit zurück. Ist nach erfolgreicher Aktivierung beider Pumpen die Feuchtigkeit wieder unter den Grenzwert gefallen, so werden die Pumpen abgeschaltet und das Tor geöffnet. Während sich das Tor öffnet ist die Lampe B eingeschaltet. Ist das Tor vollständig geöffnet, so wird die Feuchtigkeit wieder überwacht.

Modellierungsaufgaben

Durch die folgenden Schritte soll die Software strukturiert entworfen werden.

Teil 1: Spezifikation der State Machine

Überlegen Sie zunächst, welche Signale in die State Machine hinein- und welche herausgehen, bzw. welche Aktionen durchgeführt werden. Listen Sie diese Signale auf. Skizzieren Sie parallel das Systemkontextdiagramm der State Machine (Systemgrenzen und Komponenten identifizieren).

Erstellen Sie die State Machine in Magic Draw, Visual Paradigm, Visio oder auf Papier, die das oben beschriebene Steuerungsverhalten aufweist. Die Modellierung soll auf der abstrakten Ebene auf Basis Ihrer Analysen erfolgen.

Teil 2: Testfallableitung

Erstellen Sie den Zustandsübergangsbaum, der für einen Test der Implementierung der State Machine erforderlich ist.

Erstellen Sie nur für den längsten Pfad im Zustandsübergangsbaum das Sequence Diagram, welches die Interaktion aller beteiligten Komponenten darstellt. Der Sensorwert der Feuchtigkeit kann über einen Wrapper (→ Design Pattern), der im Kontext der State Machine bekannt ist, abgefragt werden. Zur besseren Übersicht erstellen Sie ein Klassendiagramm der beteiligten Komponenten.

Implementierungsaufgaben

Auf Basis Ihrer Modellierung sollen Sie das Verhalten implementieren und verifizieren. Als Basis sind die beigefügten Interfaces und Code-Fragmente zu verwenden.

Teil 3: Klassenstruktur

Für den Übergang zur Implementierung müssen Sie die State Machine realisierende Klasse in eine in eine Klassenstruktur einbetten. Analysieren Sie hierzu die Interfaces der beigefügten Sources.

Die State Machine kann auf den aktuellen Feuchtigkeitswert `humidity` direkt zugreifen. Der Zugriff erfolgt über das Interface `IHumiditySensor`.

Über weitere Interfaces sind die Methoden zum Versenden der Signale über das Netzwerk gegeben. Über weitere Methoden kann bei den Interfaces abgefragt werden, ob ein Signal empfangen wurde.

Aktualisieren Sie Ihr Klassendiagramm, welches das gesamte Softwaresystem mit der FSM darstellt, auf Basis der gegebenen Sources.

Teil 4: Teststruktur

Das Verhalten der State Machine (längster Pfad) soll mittels Unit Testing und unter Verwendung der bereitgestellten Mocks verifiziert werden.

Erstellen Sie auf Basis der Testfalleableitung den Unit-Test zur Verifikation ihres längsten Pfades. Die beigefügten Mocks müssen dabei noch mit der notwendigen Funktionalität zur Unterstützung des Tests erweitert werden. Aktualisieren Sie ggf. Ihr Klassendiagramm.

Teil 5: State Machine

Implementieren Sie die State Machine mit Hilfe des switch-case Patterns und verifizieren Sie die Implementierung gegen Ihren Test.

Bearbeitungshinweise

UML-Tool

Die Erstellung von Diagrammen kann auch auf Papier erfolgen. Für das Review im Praktikum müssen die Diagramme in einer elektronischen Form vorliegen.

Verwendung von Git

Im Rahmen dieses Praktikums soll die Verwendung eines Version Control Tools am Beispiel von Git weiter geübt werden.

Legen Sie ein Repository für das Team an, z.B. auf GitLab der Informatik, Bitbucket oder GitHub. Legen Sie die beigefügten Sources als Initiale Version dort ab.

Es sollen mindestens die Ergebnisse jeder Teilaufgabe als Commit im zentralen Repository abgelegt werden. Auch sollten die Teilaufgaben nicht nur von einer Person bearbeitet werden. Der letzte Commit auf dem Master-Branch vor dem Praktikumstermin soll in der Nachricht den Eintrag „[ready for review]“ beinhalten.