

Praktikum Software-Engineering 1 WiSe 16

Aufgabenblatt 4

Prof. Dr. Bettina Buth <buth@informatik.haw-hamburg.de>

Raum 7.86b, Tel. 040/42875-8150

Bearbeitungshinweise:

- Die Bearbeitung der Aufgaben findet in **festen Dreier- oder Viererteams** statt.
- Die bearbeiteten Lösungen werden während der Praktikumsstunde abgenommen. Dazu werden Sie abwechselnd Ihre Lösungen vorstellen.
- Es gibt **100% Anwesenheitspflicht** beim Praktikum. Beim Fehlen wegen z.B. Krankheit müssen Atteste eingereicht werden und ein Nachholtermin wird vereinbart.

Ziel des Praktikums:

- Requirements Engineering am Beispiel durchführen nach Vorgehen für Analyse und Design von RTE Software
- Use Cases identifizieren, textuell und in einem Use Case Tabellen dokumentieren
- Zustandsautomat als Dokumentation eines Teilablaufs erstellen
- Die Ergebnisse sollen stichprobenartig in der gesamten Praktikumsgruppe präsentiert werden

Vorbereitung vor dem Praktikum

- Lesen Sie die Information auf den Folien zu Kap 6a und 6b
- Lesen Sie die Information zur Agilen RTE Entwicklung
- Machen Sie sich mit den Beschreibungsmustern für AERTS vertraut, die mit der Aufgabe ausgegeben werden
- Lesen Sie die Beschreibung der Fahrstuhlsteuerung sorgfältig durch.
- Identifizieren Sie dabei Informationen, die sich auf die Steuerungssoftware beziehen und solche, die sich auf die Umgebung der Steuerung beziehen.
- Identifizieren Sie die Stakeholder für die Anwendung (speziell für Use Cases relevant!)
- Legen Sie sich für das Use Case Dokument vorab die Dokumentenrahmen an

Zusatzinformationen:

- Vorgehen für OO Analyse und Design findet sich als Foliensatz im pub unter Material zum selbständigen Arbeiten
- Gliederung für Anforderungsdokumente findet sich auf den Vorlesungsfolien
- Die Schablone für die Dokumentation von Use Cases ebenfalls in der Vorlesung und im pub
- UML Diagramme können handschriftlich erstellt werden. Alternativ können UML-Werkzeuge wie Visual Paradigm verwendet werden

Requirements Engineering: RTE-Entwicklung - Use Cases und UML

Beschreibung eine Fahrstuhl-Steuerung

Ein Auftraggeber kommt mit der folgenden Vision einer Fahrstuhlsteuerungssoftware zu Ihnen:

Das Gesamtsystem teilt sich dabei in drei Komponenten:

- die Steuerung und Überwachung der Anzeigen und Kontrollen in jedem der Stockwerke
- die Steuerung und Überwachung der Anzeigen und Kontrollen in jeder Kabine
- den zentralen Steuermechanismus, der die generelle Bewegung und Position der Fahrkabinen und Türen überwacht und steuert.

In jedem Stockwerk befinden sich

- je ein Knopf pro Fahrtrichtung für die Fahrstuhlanforderung
- Anzeigen für jeden Fahrstuhl über die aktuelle Position und Fahrtrichtung
- je zwei Sensoren pro Fahrstuhl, die oberhalb und unterhalb der Tür im Fahrstuhlschacht angebracht sind und bei gleichzeitiger Aktivierung anzeigen, dass die Fahrkabine in Position steht.
- eine Tür, die automatisch mit der Kabinentür geöffnet und geschlossen wird. Diese Tür besitzt keine eigene Steuerung, aber einen Sensor zur Überwachung, ob sie geöffnet oder geschlossen ist.

In jeder Fahrkabine befinden sich

- je ein Knopf pro Stockwerk zur Angabe des gewünschten Fahrziels
- eine Anzeige für die aktuelle Position und Fahrtrichtung
- ein Knopf für das Schnellschließen der Tür (Schließe-Knopf)
- ein Knopf für das Offenhalten der Tür (Öffne-Knopf)
- insgesamt 6 Sensoren (je drei pro Seite) in der Tür zur Erkennung von Hindernissen beim Türschließen
- ein Timer, der nach Angabe des Fahrziels oder bei vorhandenen Zieleingaben nach Öffnen der Türen als Beantwortung auf eine Anforderungen gestartet wird und bei Ablauf das Signal an die Kabinentür zum Schließen gibt; dieser Timer wird bei Betätigung des Öffne-Knopfes neu gestartet, bei Betätigung des Schließe-Knopfes gestoppt.
- ein zweiter Timer, der nach Ablauf des ersten Timers aktiviert wird. Wenn nach Ablauf des zweiten Timers die Tür nicht geschlossen ist, wird ein Fehler an die zentrale Einheit gemeldet. Der zweite Timer wird zurückgesetzt, falls die Tür wieder geöffnet wird, was an einer Reaktivierung des ersten Timers erkennbar ist.

Die zentrale Steuereinheit, deren Software Sie entwickeln sollen, hat folgende Aufgaben:

- Sie speichert und überwacht die aktuelle Position jeder Fahrkabine und stellt sie den Anzeigeelementen in den Stockwerken und Kabinen zur Verfügung. Als Input wird die Information der Positionssensoren in den Stockwerken verwendet.
- Sie kontrolliert Fahrtrichtung und Richtungsänderungen der Kabinen; Ziel ist ein bestmögliches Scheduling in Abhängigkeit von Position und Nutzeranfragen.
- Sie steuert das Öffnen und Schließen der Fahrstuhltüren in den Stockwerken
- Sie überwacht mögliches Fehlverhalten des Systems, etwa Türblockierungen.

Bei der Realisierung des Systems ist eine Reihe von Regeln zu beachten, die das Gesamtverhalten bestimmen:

- Eine Fahrkabine befindet sich im Stockwerk n wenn oberer und unterer Sensor der Stockwerkserkennung im Stockwerk n aktiviert sind. Das heißt insbesondere, dass sich der Fahrstuhl solange in diesem Stockwerk befindet solange keine andere Stockwerksposition erreicht ist.
- Die Anzeigen geben die letzte gesicherte Position der Kabinen wieder (sowohl innerhalb der Kabine als auch in den Stockwerken)

- die Türen der Kabine (und damit auch die des Stockwerks) öffnen sich ausschließlich dann, wenn sich die Kabine im zugehörigen Stockwerk befindet.
- Der Fahrstuhl kann erst dann fahren, wenn innere und äußere Tür geschlossen sind.
- Beim Schließen der Tür geht Sicherheit vor: beim Erkennen eines Hindernisses wird ein Schnellschließen ignoriert bzw. abgebrochen.
- Das Fahrverhalten soll möglichst effizient sein; d.h. dass unnötige Richtungsänderungen vermieden werden sollen und dass auf eine Anforderung aus dem Stockwerk der jeweils nähere Fahrstuhl die Anfrage annimmt. Dabei soll die erste Bedingung eine höhere Priorität haben als die zweite.

Regeln für die Türsteuerung

- Die Steuerung ist deaktiviert wenn der Fahrstuhl fährt;
- sie wird aktiviert, wenn der Fahrstuhl in einem Stockwerk ankommt - in diesem Fall ist die Tür geschlossen;
- ist die Tür geschlossen, wird sie von der Türsteuerung geöffnet;
- ist die Tür offen, so wird nach 5 Sekunden die Tür geschlossen;
- wenn beim Schließen ein Hindernis erkannt wird, wird die Tür wieder geöffnet
- wenn beim Schließen der Öffne-Knopf in der Kabine gedrückt wird, wird die Tür wieder geöffnet
- solange die Türsteuerung aktiv ist, kann die Tür durch Drücken des Öffne-Knopfes wieder geöffnet werden.
- wenn die Tür offen ist, kann sie unter Umgehung der Verzögerungszeit durch Drücken des Schnellschließ-Knopfes sofort geschlossen werden.
- die Türsteuerung wird deaktiviert wenn die Tür geschlossen ist und ein Ziel für den Fahrstuhl vorliegt.

Vorgehen allgemein:

Die Folien zum Thema RTE-Entwicklung (Vorlesung SE1 SoSe 16) enthalten eine grobe Skizze für das Vorgehen bei der Erfassung von Anforderungen für RTE-Entwicklung. Dieser Ansatz sollte bei den Aufgaben berücksichtigt werden

Aufgabe 1: Anforderungsermittlung

Bearbeitung Im Plenum; **Abnahme Aufgabe 1: im Praktikum durch Diskussion im Praktikumsplenum**

1.1 Akteure

Identifizieren Sie auf Basis des Textes die Akteure für die Anwendung Fahrstuhlsteuerung

1.2 Szenarien

Beschreiben Sie mehrere Anwendungsszenarien für die Verwendung der Fahrstuhlsteuerung

1.3 Use Cases

Ermitteln Sie die wichtigsten Anforderungsfälle für die Fahrstuhlsteuerung.
Listen Sie diese auf und stellen Sie diese in UML Use Case Tabellen auf dem Kite-Level dar.

1.4 Kontext-Modell

Identifizieren Sie auf der Basis der bisherigen Aufgaben die Nachbarsysteme **der SW-Steuerung**.

1.5 Identifizieren der wesentlichen Objekte

Identifizieren Sie auf der bisherigen Basis die Entitätsobjekte, Grenzobjekte und Steuerungsobjekte der SW-Steuerung; Dokumentieren Sie diese in einem Klassendiagramm.

Aufgabe 2: RTE Anforderungen ermitteln

Durchführung – je Team 1 Teilaufgabe; Abnahme Aufgabe 2: Diskussion im Praktikums-Plenum

2.1 Sequenzdiagramme

Beschreiben Sie mit Hilfe von Sequenzdiagrammen das Zusammenspiel der in 1.4 identifizierten Objekte für die in Aufgabe 1.3 identifizierten Use Cases

2.2 Zustandsdiagramm

Erstellen Sie zu dem Use Case zur **Steuerung der Fahrstuhltür** ein hierarchisches Zustandsdiagramm, das die Abläufe graphisch dokumentiert. Identifizieren Sie dafür zunächst die Zustände und die Ereignisse sowie Aktionen, die bei dem Übergang von einem zum anderen Zustand ausgeführt werden.

2.3 Verfeinern der Use-Cases

Durchführung: gemeinsam im Plenum; Abgabe: ein Dokument, entsprechend den obigen Vorgaben

Erstellen Sie ein verfeinerte Use Cases auf dem Wave-Level auf der Basis der Aufgaben 2.1-2.3

Hinweis: Abgabe der Use Cases und Diagramme per email an buth@informatik.haw-hamburg.de mit Betreff „[SE1] Aufgabenblatt 4“ – Kopie an alle Teammitglieder

Abgabe der schriftlichen Teile bis

Mo, 20.6.2016, 8:00

Viel Spaß!