

Praktikum Software-Engineering 1 SoSe 16

Aufgabenblatt 2

Prof. Dr. Bettina Buth <buth@informatik.haw-hamburg.de>
Raum 7.86b, Tel. 040/42875-8150

Bearbeitungshinweise

- Die Bearbeitung der Aufgaben findet in **festen Vierergruppen** statt.
- Die bearbeiteten Lösungen werden in der Regel während der Praktikumsstunde abgenommen. Dazu werden Sie abwechselnd Ihre Lösungen im Team 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:

- Arbeiten mit UML-Diagrammen, speziell State Machine Diagram
 - Lesen und analysieren von UML-Automaten
 - Entwickeln von UML-Automaten zu informellen Beschreibungen
 - Direkte Implementierung von UML Diagrammen

Vorbereitung vor dem Praktikum

- 1) Machen Sie sich mit den UML-Diagrammen vertraut – insbesondere dem State Machine Diagram
Verwenden Sie dazu auch die UML-Zusammenfassung in der Datei [uml-2-Notationsuebersicht-oose.de.pdf](#)
- 2) Lesen Sie die Aufgaben sorgfältig durch und stellen Sie Ihre Fragen zu den Aufgaben am Anfang des Praktikumstermins

Zusatzinformationen:

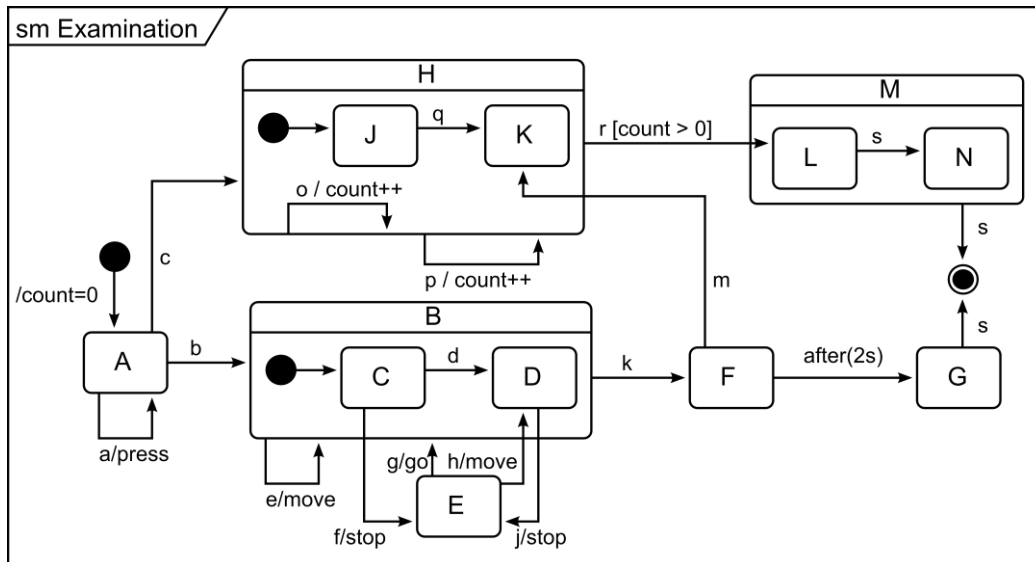
- Knappe Übersicht über UML 2.4.1: <http://www.oio.de/public/objektorientierung/uml-referenz-2-4/>

Aufgabe 1: UML-Diagramme lesen und transformieren

1.1: Hierarchische UML-Zustandsdiagramme lesen und analysieren

Abgabe: Diskussion im Praktikum, Bearbeitung 20 min

Gegeben sei das folgende Zustandsdiagramm in UML-Notation



- Vom Startzustand aus wird die Signalfolge <a, b, c> ausgelöst. In welchem Zustand befindet sich der Zustandsautomat?
- Geben Sie die Folge von Eingangssignalen an, um auf möglichst kurzem Weg (minimale Anzahl an Eingangssignalen) den Automaten zu durchlaufen und dabei folgende Zustände einzunehmen: <Start> → D → C → L → <Endzustand>

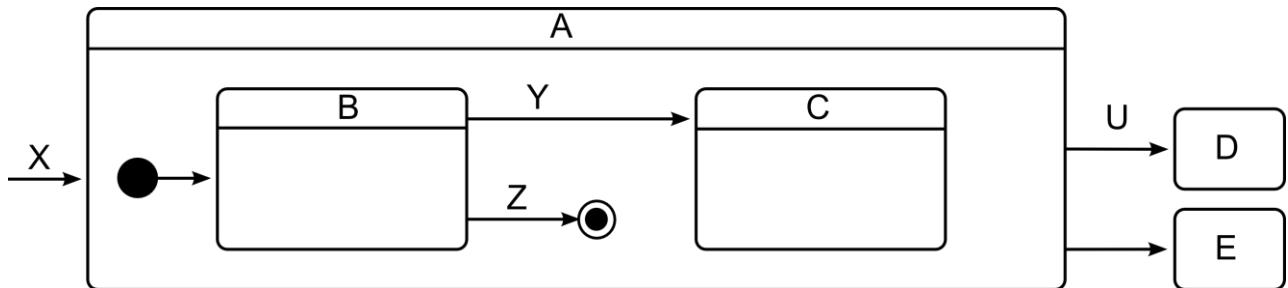
Dabei können zwischen den einzelnen Zuständen auch andere Zustände erreicht werden

- Geben Sie eine möglichst kleine Anzahl von Folgen von Events an, die zusammen alle Zustände annehmen.

1.2: Hierarchische UML-Zustandsdiagramme transformieren

Abgabe: Diskussion im Praktikum, Bearbeitung 20 min

Wandeln Sie den nachfolgenden Ausschnitt aus einer hierarchischen State Machine in eine State Machine gleicher Funktion ohne Hierarchie um.



Aufgabe 2: UML-Diagramme entwickeln und implementieren

Gegeben sei die folgende Beschreibung eines einfachen Systems zur Feuchtigkeitsregulierung:

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.

2.1: UML Zustands-Diagramm entwickeln

Abgabe: Diskussion im Praktikum, schriftliche Abgabe des Diagramms.

Entwickeln Sie ein UML Zustandsdiagramm, das der Funktion dieser Steuerung entspricht

Dokumentieren Sie dafür zunächst

- die Zustände und die Ereignisse, die zu Zustandsübergängen führen.
- Die Bedingungen und Aktionen, die bei den Übergängen relevant sind

Verwenden Sie für die Dokumentation zunächst nur Stift und Papier; präsentieren Sie Ihre Lösung auf Wandfolie (erhalten Sie im Praktikum)

2.2: UML Zustands-Diagramm implementieren

Abgabe: Vorstellung im Praktikum – Demonstration der Tests, schriftliche Abgabe des Codes

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

a) Klassenstruktur

Für den Übergang zur Implementierung müssen Sie die State Machine realisierende Klasse 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 zusätzliche Methoden kann bei den Interfaces abgefragt werden, ob ein Signal empfangen wurde.

Erstellen Sie das Klassendiagramm, welches das gesamte Softwaresystem mit der FSM darstellt.

b) Teststruktur

Das Verhalten der State Machine im Sinne des längsten Pfades soll mittels Unit Testing und unter Verwendung der bereitgestellten Stubs verifiziert werden.

Erstellen Sie einen Unit-Test zur Verifikation Ihres längsten Pfades. Die beigefügten Stubs müssen dabei noch mit der notwendigen Funktionalität zur Unterstützung des Tests erweitert werden.

c) State Machine

Implementieren Sie die State Machine mit Hilfe des Switch-Case Patterns (vgl Folien) und verifizieren Sie die Implementierung gegen Ihren Test.

Hinweis: Abgabe der Aufgabe 2 schriftlich, per email an buth@informatik.haw-hamburg.de mit Betreff „[SEP1] Aufgabenblatt 2, Aufgabe 2“ – Kopie an alle Teammitglieder

Abgabe der schriftlichen Teile bis

Do, 5.5.2016, 23:00

Viel Spaß!