

---

---

# Modellierung II – SoSe 2018 - Reflexion des State Machine Projektes - Gruppe 3

---

---

EINE KOLLISIONSFREIE ROBOTERSTEUERUNG

VERFASSER:

GRUPPE 3



ABGABETERMIN: 12.07.2018

## **16 UML Zustandsdiagramme: Zustände**

Da uns das Prinzip des Automaten mit Zuständen bereits aus Modellierung 1 bekannt war, hatten wir an dieser Stelle keine fundamentalen Schwierigkeiten.

Ein bedeutender Fehler war, den Startzustand als einen Zustand anzusehen, in den man zurückkehren kann. Außerdem hätten wir vielleicht etwas mehr mit `entrys` und der `Do`-Methode des Zustandes arbeiten sollen, etwa um das Warten auf eine Antwort darzustellen.

## **17 UML Zustandsdiagramme: Transitionen**

Die grundlegenden Möglichkeiten die zum Auslösen von Transitionen in der UML zur Verfügung stehen, bereiteten wenig Probleme. Viele mächtigere Möglichkeiten waren uns beim Lösen der Aufgabe jedoch nicht bekannt (insbesondere `when`-Statements). Hier hätten wir uns nach Verantwortlichkeiten noch tiefer in die UML einarbeiten sollen.

Ein ähnliches Problem schafften wir uns mit Verkettungen von Entscheidungen, in denen Seiteneffekte auf Variablen auftraten, die zu einem späteren Zeitpunkt als Bedingung fungierten. Hier hätten mehr Zustände geholfen.

Wir hätten uns außerdem eine detailliertere Auseinandersetzung mit solchen komplexeren Möglichkeiten im Rahmen der Vorlesung gewünscht.

## **18 UML Aktivitätsdiagramme für detaillierte Abläufe**

Da unsere Lösung ohne zusätzliche Aktivitätsdiagramme ausgekommen ist, verzichteten wir an dieser Stelle auf eine umfassende Reflexion. Das Feedback im Review hat jedoch gezeigt, dass es vermutlich an einigen Stellen sinnvoll gewesen wäre, kompliziertere Abläufe in einer Funktion zu kapseln die in einem Aktivitätsdiagramm über Selbstaufrufe dargestellt wird.

## **19 UML Klassendiagramme: Aktive Klassen und Enumerations**

Mit der Verwendung von Enumerations hatten wir in keiner Weise Probleme, da diese bereits aus der Einführungsveranstaltung in die Programmieretechnik bekannt waren. Wir empfanden sie als sinnvolle Ergänzung zu den Methodenaufrufen und dem durch den aktuellen Zustand dargestellten Zustand des Automaten.

Mit aktiven Klassen haben wir nicht gearbeitet, da unserer Meinung nach mit dem Zustandsautomaten das erwünschte Verhalten eindeutig dargestellt werden konnte.

## **20 Modellierungswerkzeuge, Teamwork und Kommunikation**

Da unser altbewährtes Tool Visual Paradigm keine besonders gute Implementation von State Machines bereitstellte haben wir uns für die Umsetzung mittels des Online-Tools `draw.io` entschieden. Da zu den

Terminen schon sehr ausgereifte Modelle vorhanden waren, beschränkte sich die Kommunikation auf die gemeinsamen Treffen, in denen wir die Modelle und Vorträge finalisieren konnten.