

3. Übungsblatt

Ausgabe: 28.05.15

Abgabe: 11.06.15

In diesem Aufgabenblatt wollen wir zum einen die Roboter um autonom agierende Roboter erweitern, und zum anderen auf eine hochperformante, verteilte, auf Aktoren basierende und in Akka implementierte Architektur umstellen.

Autonom agierende Roboter dürfen als Grundlage für ihre Steuerung nur die Daten aus dem Radar (welches das direkt sichtbare Umfeld anzeigt) und dem Langstreckenscanner (welches für die gesamte Mine die Gold- und Dilithiumvorräte anzeigt), sowie die lokale Daten (gesammeltes Gold, Energievorrat etc.) heranziehen — mit anderen Worten, genau die Daten, die auch menschliche Benutzer in ihrer Benutzerschnittstelle sehen können— und sie sollen lediglich die Kommandos absetzen dürfen, die auch der menschliche Benutzer eingibt, d.h. Bewegung in eine gegebene Richtung, einsammeln, oder pausieren.

3.1 Aktoren und Architekten

5 Punkte

Entwerfen Sie zunächst eine auf Aktoren basierende Architektur des Systems. Dazu definieren Sie:

1. Welche Aktoren gibt es?
2. Welche Nachrichten gibt es?
3. Für jeden Aktor:
 - Welche Aufgabe hat der Aktor (“Each actor does one thing”)?
 - Welche Nachrichten kann dieser Aktor verarbeiten?
 - Kann dieser Aktor abnormal terminieren, oder nicht (warum)?
 - Was passiert, wenn dieser Aktor abnormal terminiert? (wird er neu gestartet, welche Daten müssen persistiert werden?)

Das System soll (für dieses Aufgabenblatt) ohne (grafische) Benutzerschnittstelle arbeiten, d.h. es arbeiten nur autonom agierende Roboter in der Mine.

Geben Sie diese Aufgabe spätestens bis zum 03.06.2015 ab, bevor sie mit Aufgabe 3.2 beginnen!

3.2 Aktoren in Aktion

15 Punkte

Implementieren Sie die in Aufgabe 3.1 definierte Architektur. Die Architektur sollte so flexibel angelegt sein, dass autonom agierende Roboter mit verschiedenen Kontrollstrategien implementiert werden können. Beispiele für solche Kontrollstrategien sind: ein komplett zufällig agierender Roboter; ein Roboter, der soviel Gold wie möglich einsammelt; oder ein Roboter, der anfängt Energie zu sammeln, wenn sein Batteriestand zu niedrig wird. Die Funktion, die die Kontrollstrategie bestimmt, sollte als nicht vertrauenswürdig behandelt werden, und so gekapselt werden, dass sie nicht auf andere Teile (Objekte) des Systems zugreifen kann, und ihr Ausfall nicht weitere Teile des Systems in Mitleidenschaft zieht.

Implementieren Sie in ihrer Architektur drei Beispielstrategien für autonome Roboter.

Das System soll außerdem verhindern, dass die autonom agierenden Roboter beliebig schnell werden. Jedes Kommando soll daher ein bestimmtes Zeitintervall benötigen bis es vollständig ausgeführt ist. In dieser Zeit sollen keine weiteren Kommandos akzeptiert werden.

Implementieren Sie, wo sinnvoll, auch Testfälle für Ihre Aktoren.