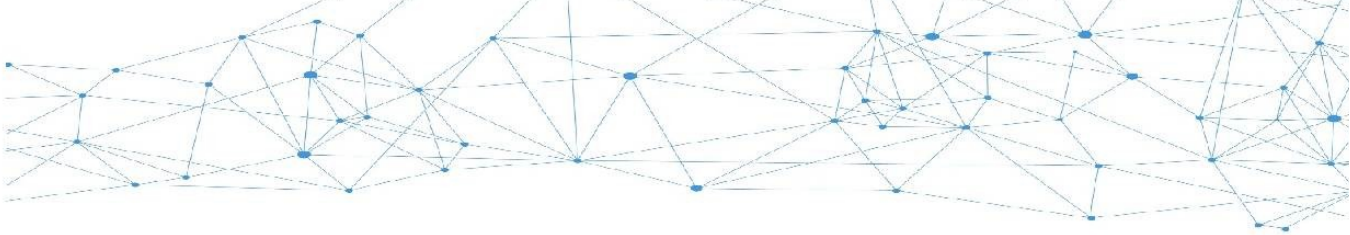
NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

Sitzung EEROS

[Sequencer]

Kalenderwoche 42



Übersicht

Anforderungen
Beispielsequenzen (Usecases)
Grundlegender Aufbau
Exception
Sequenz
Step
Probleme

Anforderungen (muss)

- [x] Mehrere Sequenzen
  - Blockierend
  - Nicht blockierend
- [ ] Exception unterbrechen Sequenzen
  - Führt Exception-Sequenz aus
  - Sequenz wird nach Exception-Sequenz neu gestartet
  - Sequenz wird nach Exception-Sequenz weiter ausgeführt
- [ ] Kommunikation mit Control- und Safety-System

Anforderungen (soll)

- Sequenzen möglichst einfach zu schreiben und abzuändern (move to P2)
- Teach-Funktionalität

Beispielsequenz A

Positionen sind bekannt

```
Open gripper
Move to P1          // home position
Move to P2          // position next to object
Move to P3          // pick position
Close gripper       // possible exception (try new position)
Move to P4          // position next to drop off point
Move to P5          // place position
Open gripper
```

Beispielsequenz B

Set speed x=3
Set speed y=2
Wait 5s
Set speed x=0
Set speed y=0

Grundlegender Aufbau

Eine Sequenz besteht aus roboterspezifischen Methoden (Steps)
Eine Sequenz kann auf eine Aufgabe (Ablauf) angepasst werden
Die möglichen Steps (Methoden) werden wärend der Entwicklung einer EEROS-Applikation definiert

Grundlegender Aufbau

Eine Sequenz besteht aus roboterspezifischen Methoden (Steps)
Eine Sequenz kann auf eine Aufgabe (Ablauf) angepasst werden
Die möglichen Steps (Methoden) werden wärend der Entwicklung einer EEROS-Applikation definiert

Exception 1/2

Exception = Interrupt

Ausgelöst von:

- Control-System
- Safety-System
- Sequence
- Benutzereingabe
- Timeout

Exception 2/2

Startet Sequenz

Laufende Sequenz wird:

- Unterbrochen
- Abgebrochen

Sequenz

Serie von Steps (mindestens ein Step)

Lesen: Control- und Safety-System

- Block im CS
- Alle “Public” Methoden und Variablen

Schreiben: Chontrol-System

- Block im CS
- Mehrere Sequenzen schreiben gleichzeitig auf einen Block?

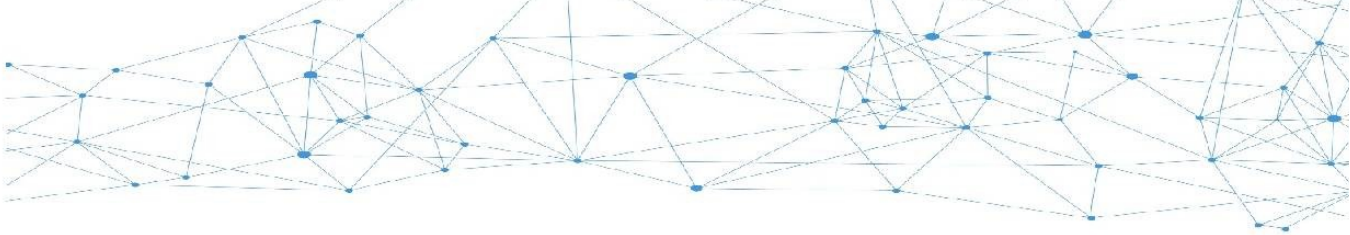
Step

Precondition = fail → Step wird übersprungen

1 Step = Eine Nachricht an CS

Postcondition:

- Step abgeschlossen (Zeit abgelaufen, Positon erreicht …)
- Exception (CS, SS, Timeout)
- Entscheidung ob:
  - Nächster Step
  - Exception Sequenz (aktuelle Sequenz schlafen legen oder abbrechen)
  - Nächste Sequenz aufgerufen wird und was die Nächste sein wird



Probleme

Zusammenführen parallel laufender Sequenzen
Parallel laufende Sequenzen schreiben die gleiche Variable im CS
Flexibilität vs. Einfachheit