

Wochenbericht VII - Praktikum "Verteilte Systeme": Implementierung MVP der Applikation und gRPC

Name Protokollführer

June 9, 2025

1 Mitglieder des Projektes

Mitglied des Projektes	Aufgabe	Fortschritt	Zeiteinsatz	Check
Manh-An David Dao	Implementierung MVP Controller	60%	3h	in progress
Jannik Schön	Implementierung MVP Model	25%	2.5h	in progress
Marc Siekmann	Implementierung MVP View	50%	4h	in progress
Philipp Patt	Implementierung gRPC	0%	0h	<ok, in review, in progress>

2 Zusammenfassung der Woche

In dieser Woche fand die Bearbeitung eines MVP der Applikation statt. Damit ist sichergestellt, dass wir eine funktionierende Applikation haben. Damit können wir Probleme, die die spätere Verteilung mit sich bringt, schneller erkennen und verstehen. Zudem wurde auch ein erster gRPC Ansatz implementiert, um eine Verteilung auf verschiedenen Programmiersprachen zu lernen.

Wesentliche pull-requests/commits sind in den Fußnoten hinterlegt.

3 Bearbeitete Themen und Schlüssel Erkenntnisse

3.1 Implementierung MVP Controller

Diese Woche wurde ein kleines Beispiel für den Controller erstellt, der Informationen an die View weiterleitet¹. Dabei wurde eine Funktionssignatur definiert (Siehe Abschnitt Implementierung MVP View), damit der Controller die View aktualisieren kann. Das Beispiel wurde zudem mit gRPC erweitert, um zu veranschaulichen, wie RPC abläuft, da in unterschiedlichen Sprachen geschrieben wird. Ziel ist es, ein frühzeitiges Grundverständnis zu schaffen und potenzielle Probleme so schnell wie möglich zu identifizieren.

3.2 Implementierung MVP Model

In dieser Woche wurde ein Wrapper für den Roboter implementiert. Ziel des Wrappers ist es, die Steuerung der einzelnen Aktoren zu kapseln und zu vereinfachen. Dabei wurde sichergestellt, dass jede gestartete Instanz des Wrappers genau einen Aktor steuert. So können mehrere Instanzen gleichzeitig betrieben werden, wobei jede Instanz unabhängig jeweils für einen spezifischen Aktor (z.B. Links/Rechts, Hoch/Runter, Vor/Zurück, Greifer) zuständig ist.

¹https://git.haw-hamburg.de/infwgi246/vs_praktikum-2025-sose/-/tree/hello-grpc-rust/Examples/grpcController?ref_type=heads

3.3 Implementierung MVP View

Es wurde festgelegt, dass das View aus den Komponenten "UI" (Anzeige) und "IO" (Steuerung) besteht. Sie haben keine gemeinsamen Schnittstellen. Damit konnte eine MVP UI für eine Browser Implementierung fertig implementiert werden ⁽²⁾. Die UI implementiert eine Schnittstelle, die von dem Controller aufgerufen wird:

```
1 updateView(AvailableRobots: String[], SelectedRobotIdx: int, Error: bool, Confirmation: bool);
```

Eine Prototyp IO auf dem ITS-Board wurde zunächst in RUST geschrieben ⁽³⁾, um die Herausforderungen zu finden, die eine Embedded-Implementierung mit sich bringt. Die RUST Implementierung wurde abgebrochen, als festgestellt wurde, dass die RPC-Kommunikation mit RUST nicht hergestellt werden konnte. Deshalb wird die IO Komponente auf dem ITS-Board zukünftig mit der Sprache C umgesetzt. Zudem wurde festgelegt, dass die UI zunächst nicht auf dem ITS-Board umgesetzt wird, da es hier nur einen Prozess gibt, sodass UI und IO synchronisiert werden müssten.

3.4 Implementierung gRPC

- embedded rpc nur mit minimal proto

4 Nächste Schritte

Dringend: Überarbeitung der arc42 Dokumentation auf den aktuellen Stand (insbesondere Schnittstellen, Komponenten der Applikation, Implementierung der Applikation (bisher nur als Skizzen ⁽⁴⁾ vorhanden))

Fertigstellung des MVP: (Festlegen Architektur-Patterns innerhalb der Applikation)

²https://git.haw-hamburg.de/infwgi246/vs_praktikum-2025-sole/-/tree/mvp/view/Application/View?ref_type=heads

³https://git.haw-hamburg.de/infwgi246/vs_praktikum-2025-sole/-/tree/hello-grpc-rust/Examples/hello-grpc-rust_itsBoard

⁴https://git.haw-hamburg.de/infwgi246/vs_praktikum-2025-sole/-/tree/notes/Notes