Wochenbericht II - Praktikum "Verteilte Systeme": Erste arc42 Iteration

Philipp Patt

May 4, 2025

1 Mitglieder des Projektes

Mitglied des Projektes	Aufgabe	Fortschritt	Zeiteinsatz	Check
Manh-An David Dao	arc42 Doku: Kapitel 1-4, 6, 11	100%	6h	ok
Jannik Schön	arc42 Doku: Kapitel 1-4, 5, 10	100%	6h	ok
Marc Siekmann	arc42 Doku: Kapitel 1-4, 8, 9	100%	8h	ok
Phillip Patt	arc42 Doku: Kapitel 1-4, 7, 12	100%	7h	ok

2 Zusammenfassung der Woche

In der zweiten Woche fand die Bearbeitung der ersten Iteration der arc42 Dokumentation statt. Wichtig war uns, dass alle das gleiche Verständnis der Aufgabe und der arc42 Dokumentation haben. Daher haben wir uns dafür entschieden, die Abschnitte 1-4 gemeinsam durchzuführen. Die nachfolgenden Kapitel haben wir, wie oben in der Tabelle beschrieben, aufgeteilt. Nach einer individuellen Einarbeitungszeit von ca. 0,5h haben wir uns die zugeteilten Abschnitte gegenseitig erklärt, sodass alle die einzelne Abschnitte inhaltlich verstanden haben. Zudem wurde geklärt in welchem Kontext der jeweilige Abschnitt im Gesamtdokument zu verstehen ist. Danach haben wir die zugeteilten Abschnitte für das bestehende Projekt ausgefüllt.

Wesentliche Pull request/commits des Projektes waren:

Erste vollständige überarbeitung arc42:
`https://git.haw-hamburg.de/infwgi246/vs_praktikum-2025-sose/-/merge_requests/5`
Wochenbericht 2:
`https://git.haw-hamburg.de/infwgi246/vs_praktikum-2025-sose/-/merge_requests/6`

3 Bearbeitete Themen und Schlüssel Erkenntnisse

Hinweis: Die Kapitelnummern beziehen sich auf die arc42 Dokumentation (https://arc42.de/overview/)

3.1 arc42 Kapitel 1 - 4

In Kapitel 1 - 4 werden die Anforderungen und die Umgebung in der das System betrieben wird beschrieben. In Kapitel 4 wird dann darauf aufbauend eine Lösungsstrategie der Architektur vorgestellt. Dies haben wir gemeinsam ausgefüllt. Wir haben festgestellt, dass wir noch unzureichendes Wissen in den meisten Themenfeldern bzgl. verteilte Systeme haben. Dieses Kapitel wird im Laufe der nächsten Wochen ausgebaut, da organisatorische Entscheidungen und Entwicklungsentscheidungen (Tools etc.) noch unklar sind. Möglicherweise entscheiden wir uns für eine tabellarische Form für Lösungsstrategien.

3.2 arc42 Kapitel 5: Bausteinsicht

In diesem Kapitel wird die Darstellung des Systems in einer Bausteinsicht erwartet. Wir haben dabei das System funktional zerlegt. Die dabei entstandenen Blöcke wurden beschrieben und in Verbindung gesetzt. Dies hilft uns dabei ein gemeinsames Verständins des Systems zu entwickeln.

3.3 arc42 Kapitel 6: Laufzeitsicht

Für die erste Iteration der Architekturbeschreibung nach arc42 verwenden wir bewusst Sequenzdiagramme, da sie sich besonders gut zur Darstellung der zeitlichen Abläufe und Interaktionen zwischen Systemkomponenten eignen – klar, verständlich und ohne unnötige Komplexität.

Andere UML-Diagrammarten wie Aktivitäts- oder Zustandsdiagramme wurden in dieser Phase bewusst ausgelassen, da sie für die grundlegende Kommunikation noch nicht erforderlich sind. Ziel ist ein einfaches, aber aussagekräftiges Bild des Systemverhaltens, das später erweitert werden kann.

3.4 arc42 Kapitel 7: Verteilungsschicht

Hier wird dokumentiert wie die Software auf verschiedenen Hardwareumgebungen gemappt wird. Häufig verwendete Umgebungen sind Entwicklungs-, Test- und Produktivumgebung. Fürs erste werden wir die gegebene Hardware (ITS-Board, RasberryPi) verwenden und von Benutzung weiterer Infrastruktur absehen, solang sie nicht notwendig wird. Auch ist für uns eine Umgebung (Produktiv) erstmal ausreichend, da noch keine tatsächliche Entwicklung oder Tests stattfinden.

3.5 arc42 Kapitel 8: Konzepte

Das Kapitel wurde als Zusammenfassung des Projektes verstanden. Somit wurden einige Oberthemen rausgesucht und anschliessend die verwendeten Konzepte beschrieben. Dies muss in den nächsten Iterationen ergänzt und verbessert werden.

3.6 arc42 Kapitel 9: Architekturentscheidungen

Das Kapitel beschreibt die Architektur Entscheidungen. Wir haben uns darauf geeinigt unsere Entscheidungen dort tabellarisch festzuhalten und mit einem Zeitstempel (Wochenzahl) zu versehen. So können wir nachschauen, wann wir eine Entscheidung getroffen haben und eine Begründung dazu. Das ergibt am Ende einen Zeitverlauf, sodass wir Architekturentscheidungen nachvollziehen können

3.7 arc42 Kapitel 10: Qualitätsszenarien

In diesem Kapitel werden die Qualitätsanforderungen des Systems beschrieben. Beim Erstellen der Qualitätsszenarien ist uns die Wichtigkeit des Verhaltens des System, während eines Stromausfalls aufgefallen. Dieses Verständins wird uns in der weiteren Entwicklung helfen.

3.8 arc42 Kapitel 11: Risiken

Da sich unser Architekturansatz noch in der frühen Phase befindet, lassen sich viele Risiken aktuell noch nicht konkret benennen. Wir konzentrieren uns daher zunächst auf technische Risiken, die sich direkt aus der gewählten Systemstruktur ergeben. Weitere Risiken werden in späteren Iterationen ergänzt, sobald zentrale Entscheidungen getroffen wurden.

4 Bezug zum Skript

- **3.1 Grundlagen für die Realisierung**
  - Für das gegeben Problem bietet sich die funktionale Zerlegung an.
    Eine Resourcen basierte Zerlegung ist Denkbar aber weniger intuitiv.
  - Solange keine guten Gründe für eine synchrone Kommunikation gegeben sind halten wir asynchrone Kommunikation für geeigneter & unkomplizierter

- Inital werden wir mit persistenter Kommunikation(TCP) designen, da Effizienzprobleme noch nicht abzusehen sind.
  in Zukunft gilt es das noch zu hinterfragen und gegen transiente(UDP) abzuwägen.
- Verwendung von RPC bieitet sich bei einer Funktionale Zerlegung an, inwiefern gRPC oder GraphQL verwendet werden steht noch aus.
- Bzgl Kopplung leigt im aktuellen Design eine "Direkte Kopplung" sowie eine "Strukturelle Kopplung" vor, da unsere Komponenten direkt miteinander kommunizieren und an den Interfaces geringfügige Einigung auf Datentypen herrschen muss
- In der GUI-Software(siehe Glossar) ist vermutlich state nicht auszuschliessen (bspw. Liste der aktiven Roboter)
  In der Roboter-Software(siehe Glossar) ist stateless Denkbar und wenn möglich zu bevorzugen.
- Das Actor Modell scheint recht Kompatibel mit unserer aktuellen Architektur.
- Im Block DHT (Gelbe Box) wird Client Server als inkompatibel mit Bidirektionaler Kommunikation definiert.
  Ist es nicht denkbar dass eine Softwarekomponente sowohl Client als auch Server implementiert?

- **3.2 Architekturparadigmen**
  - Die Architekturparadigmen "Client/Server" und "Peer-2-Peer" scheinen initial Sinvoll
  - Diese reichen für die erste Iteration und wir erweitern unser Modell nach Bedarf

5 Nächste Schritte

Da in dieser Woche die erste Iteration des arc42 Dokuments durchgeführt wurde, kann in den folgenden Wochen die Dokumentation iterativ aktualisiert werden. Zunächst haben wir aber festgestellt, dass die Anforderungen strukturierter aufgeschrieben und vervollständigt werden müssen. Die Aktualisierung des arc42 Dokuments wird erst nach der Vorlesung geschehen, da zu diesem Zeitpunkt mit den meisten neuen Erkenntnissen gerechnet wird.
Ausserdem kann als nächstes die Definition der Interfaces begonnen werden (Funktionale Zerlegung).
Weiterhin kann sich von nun an mit dem ITS-Board, Roboterarm, Toolchain und den weiteren anstehenden Aufgaben beschäftigt werden. Es wurde sich zunächst darauf geeinigt, eine Grundfunktionalität herzustellen, um die aktuelle Iteration testen zu können. Dabei könnte ein MVP helfen, der pro Iteration ergänzt wird. Die Aufgabenteilung dazu wird noch diskutiert.