

Architektur-Review

“The purpose of an architecture review is to understand the impact of every architecturally significant decision (ASD) on every architecturally significant requirement (ASR).”

Projektname	Traffic Simulation
Projektgruppe	Katzinger, Klammer, Komohorov, Mayer
Reviewgruppe	Buchegger, Crnogorcevic, Probst, Unterhitzenberger
Aufgabenstellung des Projektes laut Angabe	Entwerfen, dokumentieren und implementieren einer verteilten, komponentenbasierten Verkehrssimulation.

Inhalt

Einleitung.....	2
Rahmenbedingungen des Reviews.....	2
Fokus des Reviews.....	2
Anmerkungen.....	2
Architekturdokumentation	3
Einführung und Ziele	3
Randbedingungen	3
Bausteinsicht	3
Weitere Abschnitte	3
Designdokumentation.....	4
Gegenüberstellung von Dokumentation zu Implementierung	4
Zusammenfassung.....	5

Einleitung

Die Einleitung bietet einen Überblick über die Rahmenbedingungen als auch den Fokus dieses Reviews.

Rahmenbedingungen des Reviews

Wie mit der Projektgruppe in den Laboreinheiten vom 04.05.2017 vereinbart, dient als Grundlage für diese Review der Code- und Dokumentationsstand vom 01.06.2017.

Sowohl die Visual Studio Solution als auch die Architektur- und Designdokumentation und die darin verwendeten Abbildungen, auf der diese Review basiert, stehen als gezippter Anhang unter folgendem [Link-zum-Anhang](#), bis zum 13.09.2017, zur Verfügung. Darüber hinaus enthält der Anhang die Aufgabenstellung mit seinen gestellten Anforderungen, welche das Projekt zu erfüllen hat.

Fokus des Reviews

Dieses Dokument dient dazu, die Architektur- und Designdokumentationen der Projektgruppe auf Korrektheit zu überprüfen und mit der tatsächlichen Implementierung zu vergleichen. Besonderes Augenmerk wird dabei auf die Verständlichkeit der Dokumentation hinsichtlich der Implementierungsweise gelegt. Außerdem wird auf, sofern vorhanden, Abweichungen zwischen der Architektur als auch dem Design zur Implementierung hingewiesen und überprüft ob der Umfang der Dokumentation den nötigen Bedürfnissen entspricht.

Der Frage, ob die gewählte Architektur eine gute Wahl für die Anforderungen des Programms entspricht, wird in diesem Dokument nicht nachgegangen. Des Weiteren wird der Aufbau des Dokuments nicht bewertet.

Anmerkungen

Da das Softwareprojekt von der Projektgruppe mittels GitHub verwaltet wird, entspricht der vorliegende Stand dem Commit mit der Id 76e0ae309b8f92a67a207276b035e54e544cb7e9 vom 01.06.2017.

Die Projektdokumentation wurde dem Wiki des Github-Projektes entnommen und entspricht dem Commit mit der Id 636c6f7d3cb13b5d131d59fa63fd40774ae7ea57.

Um jedoch die offline Fähigkeit als auch die Unabhängigkeit von zukünftigen Änderungen innerhalb der Projektdokumentation zu gewährleisten, wurden Verweise durch relative Links ersetzt und die verwendeten Abbildungen der Dokumentation als offline Quellen hinzugefügt und relativ darauf verwiesen. Zur Betrachtung der offline Markdown-Files der Dokumentation in formatierter Form, empfiehlt sich der Browser Chrome unter Verwendung des Plugins MarkView.

Architekturdokumentation

In diesem Kapitel wird der Inhalt der Architekturdokumentation (Github-Wiki-File „Architekture.md“) zum Zeitpunkt der Abnahme für das Review bewertet.

Einführung und Ziele

Hier wurde zu aller erst nochmals die Aufgabenstellung („SoftwareArchitekturen2017 – Angabe.pdf“) und dessen Anforderungen festgehalten. Dieser Einstieg hilft ganz klar den Kontext des Projektes auch als Außenstehender zu verstehen. Leider stören einige Rechtschreibfehler den Lesefluss auffallend. Des Weiteren enthält dieser Abschnitt Qualitätsziele und beteiligte Stakeholder. Das zweite Anforderungsdokument („SoftwareArchitekturen2017 – Erweiterungsangabe.pdf“), mit seinen geforderten Erweiterungen an das Projekt, wurden hier noch nicht berücksichtigt.

Bei den Qualitätszielen sind die angestrebten Qualitätsmerkmale gelistet und kurz skizziert. Interessant wäre zusätzlich die Angabe des Weges oder Instrumentariums, welches zur Erreichung dieses Ziels eingesetzt wird, als auch wie diese Merkmale kontrolliert werden wollen.

Randbedingungen

Dieser Abschnitt gibt die Nicht-Funktionalen Anforderungen hinsichtlich Programmiersprache und Entwicklungsumgebung aus der gestellten Aufgabenstellung wieder, sowie die von der Projektgruppe selbst sich auferlegten organisatorischen Rahmenbedingungen. Beides ist klar und verständlich formuliert.

Bausteinsicht

Dieser Abschnitt beschreibt die einzelnen Komponenten aus denen sich die Softwarelösung architektonisch zusammensetzt. Ein Komponentendiagramm veranschaulicht gut welche Schnittstellen von welcher Komponente benötigt beziehungsweise zur Verfügung gestellt werden. Lediglich die Komponenten Process Manager und Logging bieten leider keinen Einblick wie sie in das gesamte System eingebunden werden.

Auch fehlt bei der Komponente Process Manager eine textuelle Beschreibung seiner Funktion. Alle anderen Komponenten sind verständlich umrissen und auch das Implementierungsmodell der Schnittstelle zwischen den Komponenten Simulation, UI und Traffic Light Control ist in schriftlicher Form skizziert und die Wahl begründet.

Darüber hinaus enthält dieser Abschnitt zwar eine Abbildung über das Zusammenwirken der einzelnen Komponenten, welche jedoch nicht mehr Information über das Zusammenwirken liefert als die Komponentenansicht selbst. Anzumerken ist auch hier, dass nicht hervorgeht in wie weit die beiden Komponenten Logging als auch Process Manager eingebunden werden.

Weitere Abschnitte

Zu den folgenden Abschnitten der Architekturdokumentation kann keine Aussage getroffen werden, da sie noch ohne Inhalt sind und mit „in Arbeit“ gekennzeichnet:

- Kontextabgrenzung
- Lösungsstrategie
- Laufzeitsicht
- Verteilungssicht
- Konzepte
- Entwurfsentscheidungen
- Szenarien
- Risiken und technische Schulden
- Glossar

Designdokumentation

In diesem Kapitel wird der Inhalt der Designdokumentation (Github-Wiki-File „Design.md“) zum Zeitpunkt der Abnahme für das Review bewertet.

Die vorliegende Designdokumentation besteht aus einer Überschriftenstrukturierung, stichwortartigen Besprechungsnotizen des Entwicklungsteams, sowie zwei möglichen Klassendiagrammen ohne Beschreibung. Des Weiteren wurde das Diagramm der Bausteinsicht aus der Architekturdokumentation wiederverwendet um das Design näher zu erklären, aber auch hier fehlt der schriftliche Zusammenhang, wodurch sich die Intention des Diagramms an dieser Stelle nicht erschließt.

Nachdem sich bei den Klassendiagrammen ohne weitere Erläuterungen kein Kontext herauskristallisiert, kann über die Bedeutung der einzelnen Klassen für das Gesamtsystem nur durch ihren Namen gemutmaßt werden. Vererbungsbeziehungen werden durch das Klassendiagramm veranschaulicht, weitere Informationen zu den Beziehungen der Klassen untereinander sowie über verwendete Interfaces fehlen jedoch leider.

Die weiteren Abschnitte der Designdokumentation sind noch leer, wodurch nicht mehr dazu gesagt werden kann.

Gegenüberstellung von Dokumentation zu Implementierung

In diesem Kapitel wird die Implementierung (siehe Anhang) mit der Design- und Architekturdokumentation (Github-Wiki Files „Architecture.md“ und „Design.md“) verglichen und eventuelle Abweichungen hervorgehoben.

Da die Designdokumentation zum Zeitpunkt des Reviews praktisch nur eine grobe Strukturierung ohne Inhalt darstellt, kann die Implementierung leider nicht auf die Umsetzung des Designs überprüft werden.

Die Komponentensicht der Architekturdokumentation findet sich mit vier von fünf Komponenten in der Software wieder. Diese sind die Komponenten Simulation, UI, Traffic Light Control und Logging. Letztere ist als externe Library NLog in das System eingebunden, weswegen sich hier die Frage stellt, ob das Logging als eigene Komponente in der Architekturdokumentation zu führen ist. Die Komponente Process Manager, über welche, wie bereits im Abschnitt Bausteinsicht des Kapitels Architekturdokumentation festgehalten, keine Beschreibung vorliegt, ist auch in der Software selbst nicht aufzufinden.

Ansonsten kann hier noch angemerkt werden, dass die Implementierung durch die Projektgruppe mittels Features aus Visual Studio 2017 erstellt wurde, die ein nachinstallieren von Microsoft .NET Framework 4.6.2 für Windows 10 unter Visual Studio 2015 nötig machte, sowie das Anpassen einer nicht kompatiblen Codezeile im File

„src/TrafficSimulation.Simulation.Engine/PathCalculation/QuickShortestPath.cs“. Nach dieser Anpassung ließ sich die Software einwandfrei kompilieren und ausführen.

Die laufende Software dieses Commits enthält noch keine Ampelsteuerung in der angezeigten Visualisierung und auch die Fahrzeuge können noch keine Rücksicht auf Aktionen der Komponente Traffic Light Control nehmen, da diese noch keine auszuführende Logik besitzt.

Im 4. Labor wurden zwei zusätzliche Erweiterungen als Anforderungen formuliert, einerseits das Vernetzen der verschiedenen Verkehrssimulationen der einzelnen Gruppen und andererseits der Einsatz von Hindernissen auf dem Straßennetz als zusätzliche Simulationselemente. Von diesen Erweiterungen ist erstere bereits in der vorliegenden Software umgesetzt. Beide Erweiterungen finden sich jedoch nicht in der Dokumentation als Anforderungen wieder, weshalb sie nicht näher behandelt werden.

Zusammenfassung

Sowohl die zur Verfügung stehende Implementierung des Projektes als auch dessen beiliegende Dokumentation, lassen sich als Alpha-Version beschreiben. Beides weist zum Zeitpunkt der vereinbarten Abnahme für das Review erhebliche Lücken und Unvollständigkeiten auf. So fehlt die Design-Dokumentation als auch die Implementierung der Komponente Traffic Light Control praktisch vollständig.

Aus diesem Grund ist eine tiefergehende Analyse der Software mit Kontext zur architektonischen und designspezifischen Konzeption nicht möglich.