

Bachelorarbeit Zugsimulator

Dynamische Zugstrecken importieren und visualisieren

Diplomanden



Roman Spring



Pascal Gsell

Einleitung: In dieser Bachelorarbeit wurde ein Tool geschrieben, welche Streckendaten von einer XML-Datei einliest und diese in Komponenten konvertiert. Auf diesen Komponenten kann nun ein Zug simuliert werden. Dieser löst beim Wechsel oder Überfahren von Komponenten unterschiedliche Events aus. Anhand dieser Events können spezifische Komponenten zwischengeschaltet werden, welche dann die entsprechende Hardware direkt ansteuern. Siemens Mobility AG liegt bereits eine Software vor, mit welcher sie Fehler in der Hardware verstehen und nachvollziehen können. Diese Software ist jedoch spezifisch auf das Zugsicherungssystem ZSL 90 und auf eine Teststrecke programmiert. In zwei Schritten sollen die Teststrecke austauschbar gemacht und die generischen Komponenten der Software extrahiert, neu implementiert und erweitert werden.

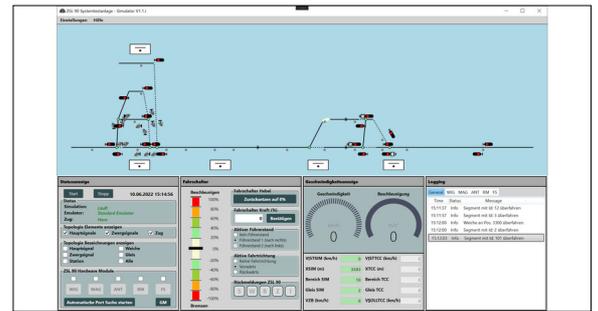
Vorgehen: Zuerst werden mögliche Formate evaluiert. Anhand einer Entscheidungsmatrix und in Absprache mit Siemens Mobility AG wird ein Format gewählt, welches umgesetzt wird. Während der Umsetzung wird auf eine gute Testabdeckung geachtet, um eine korrekte Funktionalität sicherzustellen. Anschliessend werden mögliche generische Teile evaluiert. Da sich die generischen Komponenten in drei Gruppen einteilen lassen, werden aus jeder Gruppe jeweils eine Komponente komplett entwickelt und in das bestehende Tool eingebunden. So können rasch Probleme und Hindernisse erkannt und behoben werden. Eine speziell zu erwähnende Komponente ist die "Topologie"-Komponente. Diese besteht aus vielen einzelnen Komponenten, welche zwar alle unterschiedliche Aufgaben haben, aber stark ineinander verwoben sind. Eine direkte Extraktion dieser Komponenten ist ohne ein grosses Refactoring und Clean-Up nicht möglich. Die erarbeiteten Komponenten verfügen im Track Visualizer daher über alle nötigen Funktionalitäten und können so direkt verwendet werden. In einem Demoprojekt wird die Verwendung aller Komponenten gezeigt und die bekannte ZSL 90 Systemteststrecke verwendet.

Ergebnis: Durch die Arbeit im ersten Schritt ist es nun möglich, die Strecke über eine XML-Datei zu definieren und einzulesen. Die Komponenten erhalten so die Eigenschaften aus dieser Datei. Neben den diversen kleineren Tests, um die Funktionalität sicherzustellen, wurde auch ein grosser Test geschrieben, welche die erstellten Komponenten beider Factories vergleicht.

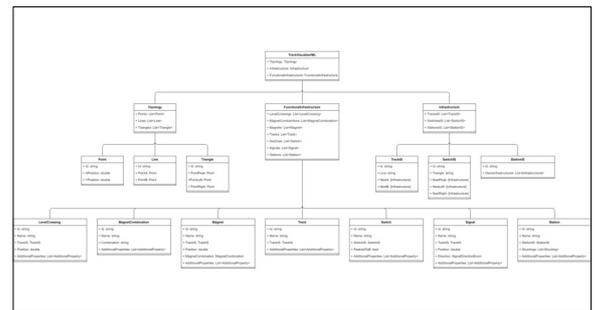
Im zweiten Schritt wurden je eine Anzeige und eine Steuerungskomponente extrahiert. Diese dienen in Zukunft als Vorlage wie die restlichen generischen Elemente extrahiert und in der bestehenden Software wieder eingegliedert werden können. Das Hauptaugenmerk im zweiten Schritt lag auf der Erweiterung der Komponenten aus dem ersten Schritt. Diese Komponenten dienen nicht mehr dem

Konvertieren, sondern stellen die Funktionalität für das Zeichnen, die Logik und die Kommunikation mit der Software bereit. Dabei wurde darauf geachtet, dass im Track Visualizer nur die Basisfunktionalitäten implementiert werden. Die Funktionalitäten welche Abhängigkeiten zum Zugsicherungssystem oder zur Software haben, wurden im Demoprojekt implementiert. Dieses Demoprojekt zeigt zum einen, dass die implementierten Funktionalitäten funktionieren und zusätzlich dient es als Ausgangslage für eine zukünftige Implementierung in die bestehende oder in eine neue Software.

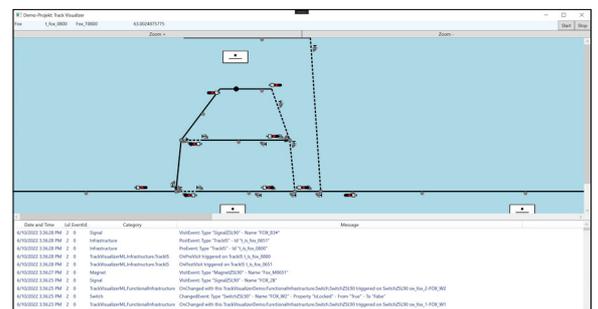
Ausgangslage: ZSL 90 Systemtestanlage der Siemens Mobility AG Eigene Darstellung



Ergebnis: Vereinfachte Übersicht der Komponenten Eigene Darstellung



Ergebnis: Demoprojekt für den Track Visualizer Eigene Darstellung



Referentin Prof. Dr. Mitra Purandare

Korreferentin Xiaolu Rao, SBB AG, Bern 65 SBB, BE

Themengebiet Software Engineering - Core Systems

Projektpartner Siemens Mobility AG, Wallisellen, ZH