

Evaluation of the SysReq¹ Controlled Natural Language for the Generation of Functional Test Cases

(Masterarbeit)



PATRICK BARAKAT

Motivation

Während eines Software-Entwicklungsprozesses werden Anforderungen meist in natürlicher Sprache erfasst. Diese können mehrdeutig, unvollständig oder widersprüchlich sein. Funktionale Testfälle auf Basis dieser Anforderungen manuell abzuleiten ist meist fehleranfällig und arbeitsintensiv.

In dem Framework NAT2TEST von G. Carvalho et. Al.¹ wird eine Methodik präsentiert um aus Anforderungen, geschrieben in einer kontrollierten natürlichen Sprache (Controlled Natural Language, CNL) namens SysReq, funktionale Testfälle zu generieren.

Stand der Technik

In der Literatur gibt es verschiedene Ansätze für die Modellierung von Anforderungen mit anschließender Generierung von Testfällen, welche sich durch die Formalisierung der Eingabe, der Ausgabe und insbesondere den zusätzlich erforderlichen manuellen Arbeitsaufwand unterscheiden². In dem Framework NAT2TEST werden signalbasierte Testfälle aus Anforderungen in SysReq-CNL mit Hilfe von intermediären Modellen (SCR und CSP) erzeugt, während zum Beispiel bei Santiago Júnior et al.³ abstrakte Testfälle mit Hilfe einer Formalisierung über Statecharts und einem benutzerdefinierten Wörterbuch generiert werden.

Zielsetzung

Im Rahmen dieser Arbeit sollen folgende Fragestellungen untersucht werden:

1. Wie sehr die Sprache SysReq-CNL geeignet ist Anforderungen aus zwei konkreten Industrieprojekten abzubilden?
2. Wie gut ist die Qualität der aus den Anforderungen generierten Testfälle im Vergleich zu einer manuell erstellten Testsuite?
3. Eignet sich der NAT2TEST-Ansatz für eines der beiden Projekte besser bzw. schlechter?

Geplante Vorgehensweise

Zu den ersten Schritten gehört die Auswahl einer Menge von Anforderungen. Diese sollen in die SysReq-DSL übersetzt werden. Dabei wird die Qualität der Testfälle im Vergleich zu manuell erstellten Testfällen mittels Mutation Based Testing beurteilt.

¹ Carvalho, Gustavo et al. (2015): NAT2TEST tool: From natural language requirements to test cases based on CSP. In: Software Engineering and Formal Methods: Springer, S. 283–290.

² Colombo, Christian et al. (2014): Verifying Web Applications: From Business Level Specifications to Automated Model-Based Testing. In: *Electron. Proc. Theor. Comput. Sci.* 141 (3), S. 14–28. DOI: 10.4204/EPTCS.141.2.

³ Santiago Júnior et al. (2012): Generating model-based test cases from natural language requirements for space application software. In: *Software Qual J* 20 (1), S. 77–143.