

Grafisch änderbare visuelle Testfallspezifikation in Arttest

(Bachelorarbeit)



JONATHAN MÜLLER

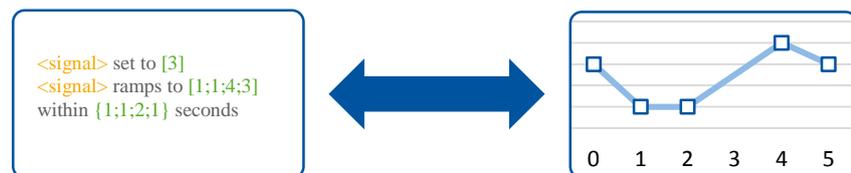
Motivation

Zur Spezifikation von Testfällen für Simulink-Modelle werden die Verläufe der Eingangssignale und die erwarteten Verläufe der Ausgangssignale benötigt. Das Programm Arttest, das am Lehrstuhl Informatik 11 entwickelt wird, ermöglicht die Spezifikation von Signalverläufen mithilfe einer eigens dafür entwickelte domänenspezifischen Sprache. Außerdem werden die daraus resultierenden Signalverläufe grafisch visualisiert. Das textuelle Eingabeformat, das zur Spezifikation verwendet wird, setzt die Kenntnis der verwendeten Sprache voraus und ist für eine zeitliche Übersicht nicht gut geeignet. Die aus der Spezifikation generierten Signalverläufe sind für den Anwender oft einfacher zu verstehen. Eine direkte Manipulation dieses Formats würde das Erstellen von Testfällen unterstützen und vereinfachen, sowie fehlerresistenter machen.

Zielsetzung

Es soll ein Editor zur grafischen Bearbeitung von Testfällen innerhalb von Arttest entwickelt werden.

Der Editor soll die Signalverläufe eines Testfalls darstellen und per Drag-and-Drop intuitiv bearbeitbar machen. Dafür soll eine Palette mit vordefinierten Arten von Segmenten (wie Sprungfunktionen, Rampen usw.) bereitgestellt werden. Die Parameter der einzelnen Segmente sollen durch verschiebbare Kontrollpunkte manipuliert werden können. Beispielsweise soll die Sprungfunktion einen Kontrollpunkt enthalten, der sowohl den Zeitpunkt als auch den Wert des Signals verändert. Bei jeglichen Änderungen soll die textuelle Spezifikation an die Änderungen angepasst werden. Umgekehrt soll der visuelle Editor aber auch auf Änderungen in der textuellen Spezifikation reagieren, sodass der textuelle und der grafische Editor stets synchronisiert sind.



Stand der Technik

Im Rahmen einer vorherigen Masterarbeit wurde ein visueller Editor entwickelt, der es erlaubt Sprachelemente anhand einer Zeitleiste anzuordnen. Im Gegensatz dazu zielt diese Arbeit darauf ab, die aus der Sprache generierten Signale direkt zu manipulieren. Um den grafischen Editor zu realisieren wird GEF 4¹ und/oder JavaFX zum Einsatz kommen. Bei GEF 4 handelt es sich um ein Eclipse-Framework, das bereits einige Editorfunktionen wie Paletten bereitstellt.

Geplante Vorgehensweise

Nach der Entwicklung eines Konzeptes zur Synchronisation von Text- und visuellem Editor, sowie der Entwicklung eines User Interfaces, soll der Editor implementiert werden. Speziell soll dabei auf intuitive Bedienbarkeit und verständliche Darstellung der Signale geachtet werden.

¹ <https://wiki.eclipse.org/GEF/GEF4>