

Statisches Typen von JavaScript-Programmen

Phillip Heidegger

Der Internetuser ist in den letzten Jahren deutlich anspruchsvoller geworden. Als Folge reicht es nicht mehr aus, auf eine HTTP Anfrage mit statischem (X)HTML zu reagieren. Vielmehr ist es heute üblich dass Webseiten Ihre Daten dynamisch per HTTP Requests vom Server anfragen, falls diese gebraucht werden. Um dieses Verhalten zu programmieren gibt es zwei Ansätze. Zuerst einmal ist es möglich mit Webtoolkits wie z.B. GWT (Google Web Toolkit) JavaScript Code zu generieren, oder der entsprechende JavaScript Code wird von Hand vom Programmierer geschrieben. Der erste Ansatz bietet eine deutlich höhere Zuverlässigkeit, da eine große Anzahl Fehler durch die richtige Wahl der Quellsprache vermieden werden können. Leider geht durch den Einsatz eines solchen Toolkits häufig ein Teil der Flexibilität verloren. Falls dies notwendig ist, muss der Programmierer auf JavaScript zurückgreifen.

JavaScript Programme statisch zu analysieren ist wegen vieler Eigenschaften der Programmiersprache schwierig. Im Vortrag werden zuerst kurz anhand von einigen wenigen Beispielen typische JavaScript Programmierfehler aufgezeigt.

Anschließend wird ein Typesystem vorgestellt, dass für eine Teilmenge von JavaScript analysiert und dennoch übliche Muster der JavaScript Programmierer ermöglicht.

Progress und Preservation sind für das Typsystem bewiesen. Das Erstellen eines Inverenzalgorithmus ist zur Zeit in Arbeit und steht als Diskussionsthema zur Verfügung. Ziel der Inverenz ist es, JavaScript Code mit möglichst wenig Annotationen versehen zu müssen, und so die Migrationsarbeit von ungetypten JavaScript Code zu minimieren.