

Zusammenfassung der Dissertation „Kontinuierliche Qualitätskontrolle von Webanwendungen auf Basis maschinengelernter Modelle“

Stephan Windmüller

27. Mai 2014

Das Internet ist in den letzten Jahren weiter rasant gewachsen, was zu einer Reihe von neuentwickelten Webanwendungen führte. Diverse Programme, die ursprünglich für den Desktop entwickelt wurden, sind jetzt auch in einer Variante verfügbar, für die nur ein Webbrowser notwendig ist. Dazu gehören beispielsweise E-Mail-Clients, neuerdings jedoch auch vollständige Office-Lösungen.

Ein generelles Problem der Softwareentwicklung, die Durchführung von Anwendungstests zur Qualitätskontrolle, besteht jedoch auch bei Webanwendungen weiterhin. Verschärft wird es durch den Umstand, dass bei einer solchen Anwendung alle Nutzer automatisch die aktuelle Version benutzen und keine Möglichkeit haben, bei Problemen auf eine ältere Revision zu wechseln.

In der Dissertation „Kontinuierliche Qualitätskontrolle von Webanwendungen auf Basis maschinengelernter Modelle“ wird daher ein Ansatz gezeigt, der über einen kontinuierlichen Lernvorgang in der Lage ist, Änderungen im Systemverhalten aufzudecken. Dazu werden vier verschiedene Herangehensweisen vorgestellt, die auch kombinierbar sind.

Kontinuierliches Lernen

Mit dieser Methode wird die aktuelle Entwicklungsversion dauerhaft durch einen Lernalgorithmus untersucht und aus den Ergebnissen ein Modell des Systems erzeugt. Zwar stellt dieses Modell nur eine Approximation des tatsächlichen Systemverhaltens dar, die kontinuierliche Komponente des gezeigten Ansatzes sorgt jedoch für eine stetig verbesserte Genauigkeit.

Ist eine neue Version verfügbar, so wird sie zunächst darauf überprüft, ob Veränderungen gegenüber dem Vorgänger vorhanden sind. Treten Unterschiede auf, so lässt sich

die Ursache auf diese Weise schnell eingrenzen. Andernfalls können die zuvor gelernten Informationen weiterverwendet werden, um das Modell weiter zu verbessern.

Häufig kommt es vor, dass ein Fehler keine direkt sichtbaren Auswirkungen hat und daher erst nach einiger Zeit entdeckt wird. Ist dann die Ursache nicht sofort klar, kann es hilfreich sein, die spezifische Änderung zu finden, welche das Problem verursacht hat. Die für das kontinuierliche Lernen notwendigen Anpassungen erlauben in diesem Fall eine effiziente Rückwärtssuche in früheren Systemversionen. In Kombination mit einem Versionskontrollsystem lässt sich dann ermitteln, wer die Änderung durchgeführt hat und was der Grund für diese war.

Vergleich des Verhaltens von Präsentations- und Steuerungsschicht

Webanwendungen sind häufig in mehreren Schichten aufgebaut. Eine Aktion in der Präsentationsschicht durch den Nutzer wird durch eine Aktion in der darunterliegenden Steuerungsschicht ausgeführt. Durch die Vielzahl an Endgeräten kann es jedoch geschehen, dass mehrere Präsentationsschichten gleichzeitig auf dieselbe Steuerungsschicht zugreifen, weshalb vor allem Schutzmaßnahmen auf dieser unteren Schicht eingebaut sein sollten. Ein Vergleich der Schichten ermöglicht es, Unterschiede im Verhalten festzustellen.

Verifikation von Systemmigrationen

Im Laufe der Entwicklung einer Webanwendung werden häufig neue Funktionen hinzugefügt, die eine Anpassung der Persistenzschicht benötigen. An dieser Stelle müssen bereits im System existierende Daten in das neue Format überführt werden, sodass sich das Verhalten gegenüber neu hinzugefügten Daten nicht unterscheidet. Eine automatische Untersuchung, die Zustände von der alten Version übernimmt und nach der Migration weiterverarbeitet, kann dabei helfen, Probleme bei einer solchen Migration aufzudecken.

Zusätzlich zu diesen vier Verfahren wurde die Konstruktion von automatisierten Testfällen vereinfacht. Durch die Kombination der kontinuierlichen Qualitätskontrolle mit einem Werkzeug zur graphischen Prozessmodellierung ist eine Spezifikation von Testfällen auch für Personen möglich, die über wenig oder keine Programmierkenntnisse verfügen.