

WEBANWENDUNGEN ERFOLGREICH TESTEN

Paper des Monats: „Analysis And Testing Of Web Applications“

In unserem „Paper des Monats“ September stellen wir diesmal einen Beitrag aus dem Jahr 2001 vor, der dieses Jahr mit dem ICSE Award „Most influential Paper“ ausgezeichnet wurde. Hervorgehoben wurde eine Pionierarbeit zum automatisierten Testen von Webanwendungen und Webseiten. Während Entwickler in den Bereichen traditioneller Software bereits auf Modelle und Testverfahren der Qualitätssicherung zurückgreifen konnten, hatte diese Möglichkeit im Bereich der Webanwendungen bis dahin gefehlt. Die Autoren Filippo Ricca und Paolo Tonella beschreiben in ihrer Arbeit „Analysis and Testing of Web Applications“ Herausforderungen für das Testen und Analysieren von Internetanwendungen und stellen die Entwicklung der Serviceprogramme „ReWeb“ und „TestWeb“, die automatisierte Testverfahren erzeugen können, vor.

Web Anwendungen gestalten sich vielfältig

Während Test- und Analyseverfahren bereits fester Bestandteil der traditionellen Softwareentwicklung waren und durch allgemein anerkannte Regeln und Abläufe definiert wurden, fehlten derartige Me-

thoden für Webanwendungen. Durch die rapide Entwicklung von Funktionen und Anwendungen des Internets und der damit einhergehenden Professionalisierung dieses Bereiches nahm der Bedarf an Methoden der Qualitätssicherung zu. Eine Herausforderung sahen die Forscher in den unterschiedlichen Ausprägungen von Internetseiten, die sowohl als statische als auch dynamische Seiten auftreten oder Links, Frames und Formulare beinhalten können. Die Forscher übertrugen Analyse- und Testmethoden der traditionellen Softwarequalitätssicherung auf Webanwendungen und betonten gleichzeitig die unterschiedlichen Anforderungen der Systeme.

Analysemodelle für Webanwendungen

Um Webanwendungen umfassend zu analysieren und zu testen ist die Erstellung eines UML-Modells für Ricca und Tonella eine Grundvoraussetzung. Dafür haben die Autoren das UML-Metamodell einer gewöhnlichen Webapplikationsstruktur erstellt, bei dem Interaktions- und Navigationsmodelle hervorgehoben wurden und Architekturperspektiven zweitrangig waren.

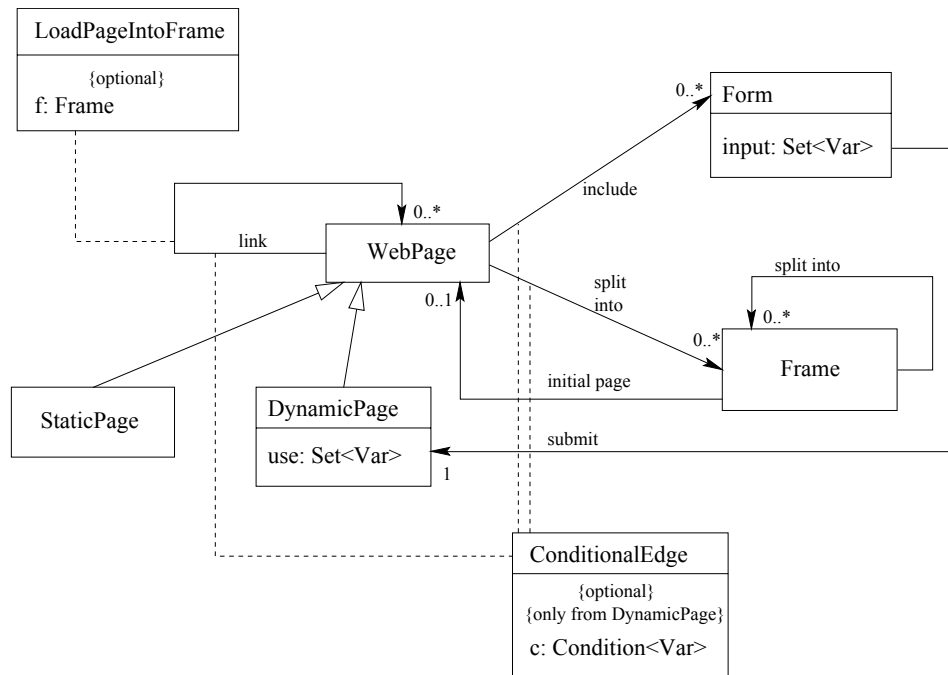


Abbildung 1: Metamodell einer generischen Webseitenstruktur. Das Modell ist eine Umschreibung einer vorgegebenen Seite. (aus Ricca und Tonella, 2001)

Das zentrale Objekt in diesem Modell ist die Webseite, die Informationen enthält, die für den Nutzer angezeigt werden und andere Seiten durch Navigationsverbindungen erreichen kann. Das Modell bildet auch dynamische Elemente ab. Zum Zeitpunkt der Veröffentlichung wurde die Dynamik vornehmlich durch Frames und Formulare geboten, daher haben die Autoren gezeigt, wie UML Modelle einer Frameseite und einer Formularseite aussehen können:

Testen von Web-Applikationen

Das Testen von Webanwendungen dient der Qualitätssicherung von Software und soll sicherstellen, dass Faktoren der Funktionalität, Zuverlässigkeit, Benutzbarkeit, Effizienz sowie der Änderbarkeit und Übertragbarkeit von Softwaresystemen gewährleistet sind. Dabei werden verschiedene Arten von Tests angewendet, die in Abhängigkeit der Beschaffenheit einer Anwendung gestaltet werden. Grundsätzlich wird eine Unterscheidung zwischen

statischen und dynamischen Verifikationen getroffen, wobei der Unterschied darin besteht, dass dynamische Tests das System in der Ausführung testen und statische Tests sich zum Beispiel mit der korrekten Beschreibung im Benutzerhandbuch oder der Rechtschreibprüfung im Quelltext befassen. Die Autoren haben verschiedene statische und dynamische Testverfahren beschrieben, die für Webapplikationen erforderlich sind:

Statische Verfahren:

Statische Verfahren untersuchen Webanwendungen auf Fehler und Anomalien, wie zum Beispiel der Aufdeckung ungültiger Navigationswege, unerreichbarer oder nicht existierender Seiten. Außerdem können damit Probleme des Ladens und der Zuordnung von Frames aufgedeckt werden. Mit der statischen Analyse werden Datenabhängigkeiten getestet und Abläufe analysiert, wodurch der Informationsfluss der Anwendung dargestellt werden kann und falsch definierte Variablen, die

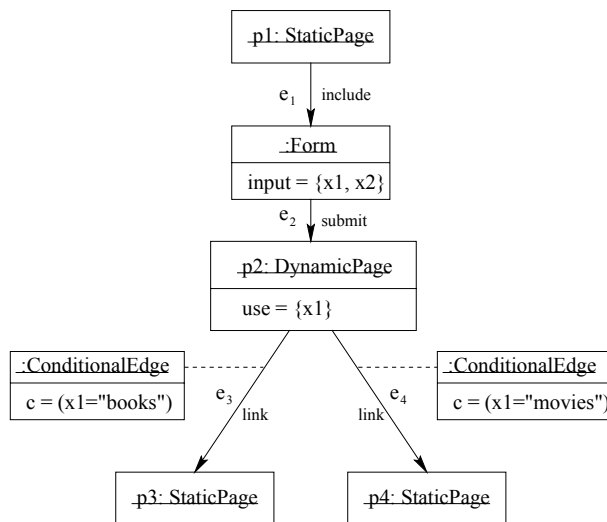


Abbildung 2: Beispiel eines UML-Modells einer Webseite, die Formulare beinhaltet. (aus Ricca und Tonella, 2001)

zum Beispiel unerwünschte Zustände erzeugen, erkannt werden können. Für Webanwendungen ist es außerdem wichtig zu testen, ob sogenannte „Dominators“ funktionieren. Das sind Seiten die Warnmeldungen oder AGBs enthalten und deswegen bestimmten Inhalten vorangestellt werden müssen. Auch die Benutzerfreundlichkeit einer Seite kann anhand der „shortest path“-Analyse überprüft werden, da untersucht wird, wie viele Seiten ein Benutzer aufrufen muss, um das gewünschte Ziel zu erreichen.

White-Box-Verfahren:

Die dynamische Analyse beschränkt sich auf Methoden des White-Box-Testens, bei denen die interne Struktur von Webanwendungen überprüft wird. Hierfür werden zum Beispiel der Quellcode eines Systems und interne Programmstrukturen auf Fehler untersucht, indem jede Anweisung einmal oder mehrmals durchlaufen wird. Das Testen von Webapplikationen verläuft zum Beispiel als Testlauf einer Sequenz von Seiten, an deren Ende das Ausfüllen eines Formulars steht. Dieser Verlauf wird in Form einer URL-Sequenz dargestellt, wel-

che die angefragten Seiten hervorhebt und diese wenn nötig mit Eingabevariablen versieht. Die ausgegebenen Seiten werden gespeichert, woraufhin der Output mit dem Soll-Zustand verglichen wird. Der Vorteil gegenüber traditioneller Software ist, dass die Auswahl des zu testenden Zweigs meist direkt über die Eingabe des Hyperlinks eingepflegt werden kann und nicht vordefiniert werden muss. Weitere Testkriterien für Webapplikationen sind:

- „Page testing“: Jede Seite wird mindestens einmal aufgerufen.
- „Hyperlinking testing“: Alle Hyperlinks jeder Seite werden mindestens einmal durchlaufen.
- „Definition-use testing“: Alle Navigationspfade, welche die zu testende Variable beeinflusst, werden durchlaufen und zwar jeweils einmal für jede mögliche Ausprägungen der Variable.
- „All-uses testing“: Mit jeder möglichen Ausprägung der zu testenden Variable wird mindestens ein abhängiger Navigationspfad durchlaufen.
- „All-path-testing“: Jeder Pfad einer Seite wird in verschiedenen Testfällen mindestens einmal durchlaufen.

Ricca und Tonella haben sich bei der dynamischen Analyse auf die „problematischen“ Seiten konzentriert und statische Seiten zu zweitangigen Einheiten erklärt, da diese unveränderliche Informationen beinhalten.

Das Erzeugen von Testfällen

Um die Bedingungen für White-Box-Tests zu erfüllen wird ein Satz von Pfaden im Webseitengraph ausgewählt, der Eingangswerte beinhaltet, jedoch nur automatisiert durchgeführt werden kann, wenn keine Eingangswerte oder „conditional edges“ berücksichtigt werden müssen. Falls dieser Umstand

berücksichtigt werden muss, werden die Werte manuell eingegeben und können dann automatisch ausgewertet werden.

ReWeb und TestWeb

Die Forscher entwickelten die Programme ReWeb und TestWeb, um das Testen von Webanwendungen zu unterstützen.

Das **ReWeb** besteht aus den Modulen „Spider“, „Analyser“ und „Viewer“. „Spider“ lädt alle anvisierten Seiten herunter, unterscheidet dabei aber nicht zwischen statischen und dynamischen Seiten, es sei denn, die Seite ist nur über ein Formular erreichbar. Der Anwender kann dynamische Seiten manuell kenntlich machen und mit Eingabewerten versehen. Danach erstellt das Programm aus den heruntergeladenen Seiten ein UML-Modell. Das Modul „Analyser“ benutzt das UML-Modell, um die oben genannten statischen Analysen durchzuführen. Das Modul „Viewer“ visualisiert das UML-Modell und die Ergebnisse der Analysen als „Graphical User Interface“ (GUI), welches zum Beispiel ermöglicht, einzelne Elemente zu vergrößern oder mithilfe der Suchfunktion zu finden. Außerdem werden verschiedene Übersichten generiert: Die „history view“, welche die Entwicklung der Seite unter zeitlichen Aspekten festhält, die „system view“, welche

die Seiten in einer Verzeichnisstruktur darstellt und die „data flow view“, die lesende und schreibende Zugriffe der Seiten auf Variablen ausgibt.

Die Aufgabe des **TestWeb** Programms umfasst die Festlegung einer Auswahl von Pfaden aus dem UML-Modell, um daraus Testfälle für das ReWeb zu generieren. Erzeugte Testfälle sind UML-Sequenzen, welche nach einmaliger Durchführung die Erfassung ausgewählter Kriterien erlauben. Die Werte müssen an dieser Stelle manuell eingegeben werden und können dann dem Webserver URL-Anfragesequenzen jedes Testfalls mit den geeigneten Eingabewerten übermitteln. Die vom Server zurückgelieferten Seiten werden gespeichert, und der Anwender überprüft dann, ob die ausgegebenen Werte zu dem eingegebenen Wert passen.

Fazit:

Das Paper „Analysis and Testing of Web Applications“ gab wichtige Impulse für das Testen verteilter Anwendungen. Die Beiträge zur Generierung von Testdaten und von Testmodellen haben die Forschung in diesem Bereich vorangebracht und speziell auch spätere Forschungen zum Testen von sogenannten „Web-2.0-Anwendungen“ (genauer gesagt asynchrone Web-Technologien wie z.B. AJAX) stark beeinflusst. Sie dienen als Grundlage

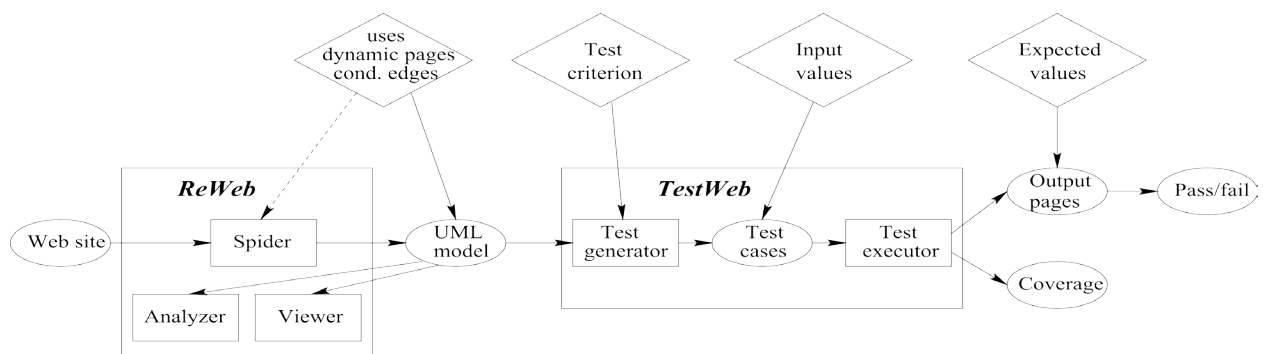


Abbildung 3: Die Darstellung von ReWeb und TestWeb während des Analyse- und Testprozesses (aus Ricca und Tonella, 2001)

für eine ganze Reihe weiterer Veröffentlichungen. Die Bandbreite reicht von Beiträgen zu Softwaretests über Reverse-Engineering bis hin zur automatischen Erkennung von sogenannten „Clickjacking-Angriffen“. Der besondere Einfluss des Papers beruht auch auf der stetig wachsenden Bedeutung verteilter, webbasierter Anwendungen.

Quelle:

Filippo Ricca und Paolo Tonella: Analysis and Testing of Web Applications, In Proceedings of the 23rd International Conference on Software Engineering (ICSE'01), IEEE Computer Society, Washington, DC, USA, 25-34. 2001. ISBN 0-7695-1050-7.

<http://dl.acm.org/citation.cfm?id=381473.381476>

Autor: Michaela Trebing