

Prozeßschritte zur Testfallauswahl bei der Testfallgenerierung aus UML-Modellen

Matthias Hudler, Michael Krüger

Fraunhofer-Einrichtung Systeme der Kommunikationstechnik
Hansastraße 32
80686 München
{matthias.hudler | michael.krueger}@esk.fraunhofer.de

Abstract: Mit Hilfe des vorgestellten Ansatzes soll eine Testfallauswahl im Rahmen einer automatisierten und modellbasierten Testfallgenerierung ermöglicht werden. Dazu wird das UML-Systemmodell entsprechend erweitert und ein mögliches Verfahren zur weiteren Informationsverarbeitung vorgestellt. Diese Testfallauswahl ist ein Teilprozeß eines Verfahrens zur Testfallgenerierung, welches aus UML-Modellen passende Testfälle in der gewählten Zielsprache mit Unterstützung einer Tool-Chain ableitet. In diesem Papier wird beschrieben, wie auf Basis von UML-Systemmodellen eine risikobasierte Testfallauswahl ermöglicht wird.

1 Ausgangssituation und Motivation

Die gezielte Auswahl von Testfällen spielt insbesondere bei Regressionstests - speziell bei einer automatisierten maschinellen Generierung einer Vielzahl von Testfällen - eine große Rolle. Dabei sollte eine Auswahl anhand von verschiedensten Kriterien geeignete Testfälle identifizieren bzw. priorisieren und gleichzeitig die Fehlerrate im Vergleich zur Durchführung aller Testfälle möglichst konstant halten. Da man bei den hier betrachteten Black-Box-Tests keinen Einblick in den Aufbau der Programme hat, ist man bei der Auswahl auf bestimmte ergänzende Angaben angewiesen. Dabei soll die Testfallauswahl als Teilablauf in einen Gesamtprozeß einer Testfallgenerierung mit Hilfe von UML-Modellen eingebettet sein und diesen entsprechend ergänzen. Als problematisch kristallisiert sich hier die Fragestellung heraus, wie und vor allem welche Informationen, die die Datenbasis in Form von Kennzahlen für eine, mögliche risikobasierte (e.g. Risiko = Schadensausmaß x Eintrittswahrscheinlichkeit), Auswahl am Ende des Prozesses darstellen, in das UML-Modell bzw. in die einzelnen Diagramme eingebracht werden können. Weiters ist zu klären, wie die Informationen geeignet interpretiert werden können und der Zugriff auf diese in dem Generierungsprozeß erfolgt. Schließlich ist noch der Zeitpunkt der Auswahl zu diskutieren, bevor die durchzuführenden Testfälle automatisch mit Hilfe eines risikobasierten Verfahrens identifiziert werden. [vgl. beispielsweise Am05, CPS02, Ot99, Re04]. Hier stellen wir einen Ansatz zur Lösung der oben besprochenen Fragestellungen vor. Dabei stützen sich

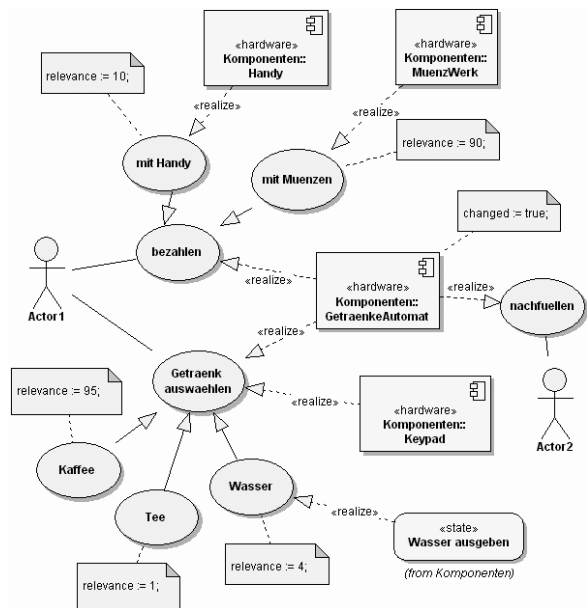
die folgenden Überlegungen auf die Vorarbeiten und den erarbeiteten Rahmen der korrespondierenden Testfallgenerierung in dem Forschungsprojekt. [Kr07]

2 Erweiterung des UML-Modells zum Zwecke der Testfallauswahl

Um eine Testfallauswahl durchführen zu können, sind zusätzliche Informationen über das zu testende System nötig, welche im Systemmodell hinterlegt werden. Hierzu verwenden wir das allgemeine Konstrukt der *Annotationen* aus UML [Om05]. Dies minimiert den Aufwand, weil zum einen keine Erweiterung des UML-Metamodells notwendig ist und zum anderen die bereits aus dem Entwicklungsprozeß vorhandenen Modelle verwendet werden können. *Annotationen* sind eine Art von Notizen, in denen der Modellierer beliebige Informationen hinterlegen kann, die vom Werkzeug in das XMI-Austauschformat übernommen werden. Sie können in jeder Art von UML-Diagrammen verwendet werden, was sie sehr flexibel einsetzbar macht. Allerdings stützt sich die Testfallgenerierung in dem hier vorgestellten Ansatz derzeit auf Zustands- und Klassendiagramme, ergänzend dazu werden Use-Case-Diagramme bei der Testfallauswahl verwendet. Die Abbildung zeigt ein Use-Case-Diagramm eines Getränkeautomaten, in dem Informationen zur Testfallauswahl in Form von *Annotationen* hinzugefügt wurden.

Um nun diese Informationen später geeignet weiter verarbeiten zu können, müssen eine entsprechende Grammatik für die zugrunde liegende Sprache definiert und die Angaben hinsichtlich dieser Syntax korrekt formuliert werden. Diese Grammatik wurde bereits ausgearbeitet, jedoch ist sie aus Platzgründen hier nicht beschrieben.

Damit eine Testfallauswahl geeignet durchgeführt werden kann, muß bekannt sein, welche Teile des UML-Modells von den einzelnen Testfällen abgedeckt werden. Der Inhalt der *Annotationen* wird im Zuge der Testfallgenerierung nicht in die entstehenden Testfälle übernommen. Deshalb muß die Testfallauswahl selbst die *Annotationen* aus dem Modell mit Hilfe von IDs lesen. Da Use-Cases, wie schon erwähnt, eine Erweiterung für die Testfallauswahl darstellen, decken die generierten Testfälle immer nur Elemente der Zustands- und Klassendiagramme ab. Aus diesem Grund müssen sie mit den Elementen des Use-Case-Diagramms in Beziehung gesetzt werden. Dazu werden im Use-Case-Diagramm



Elemente aus den übrigen Diagrammen integriert und mit den passenden Elementen aus dem Use-Case verbunden. Für die Verbindung wurde der Stereotyp <<realize>> gewählt, welcher uns, in dem verwendeten Modellierungswerkzeug, am Besten geeignet schien. Es kann jedoch auch ein anderer Stereotyp verwendet werden, da er in der Weiterverarbeitung an dieser Stelle keine Rolle spielt. Wichtig ist die Verbindung an sich. Die Abbildung zeigt, wie Elemente des Use-Case-Diagramms mit Elementen aus anderen Diagrammen verbunden wurden; z.B. wird der Vorgang „Bezahlen mit Münzen“ durch die Hardware-Komponente „Münzwerk“ realisiert, die aus einem nicht dargestellten Komponentendiagramm stammt.

Derzeit werden drei Auswahlkriterien verwendet, deren jeweilige Ausprägung in den *Annotationen* hinterlegt wird. Die drei Auswahlkriterien sind:

- **Gebrauchshäufigkeiten (*relevance*)** Die Idee hierbei ist, daß man geschätzte oder besser empirisch eruierte Häufigkeiten der Benutzung durch den User anbringt. Mit Hilfe dieser Informationen lassen sich für die Testfallauswahl gewisse Präferenzen, die aus der praktischen Verwendung des Produkts bzw. Systems entstehen, ableiten. Dabei muß nicht die komplette Information über die Verteilung vorliegen (also in der Summe genau 100% ergeben), sondern es muß lediglich überprüft werden, daß die Summe nicht 100% übersteigt.
- **Angaben über Wertigkeiten** Hier bringt nun beispielsweise der Testingenieur oder die Produktentwicklung ihre Einschätzung über die Wertigkeit von Vorgängen zum Ausdruck. Dies geschieht in dem hier gemachten Vorschlag in zwei Wertigkeiten: „*important*“ und „*most important*“. So lassen sich als wichtig eingeschätzte Aktivitäten markieren. Dies kann zum Beispiel auf den bisherig gemachten Erfahrungen mit diesen Funktionalitäten oder einer postulierten Relevanz fußen.
- **Geänderte oder neue Komponenten (*changed*)** Das letzte Kriterium bezieht sich auf die Kennzeichnung von geänderten oder neuen Komponenten im System. Dadurch werden indirekt diejenigen Testabläufe gekennzeichnet, die die abgeänderten oder neuen Teile des Systems testen sollen. Dies ist wichtig, um eine entsprechende Testfallauswahl bei Regressionstests zu leisten.

Weitere Kriterien sind denkbar, jedoch werden in einem ersten Ansatz die oben beschriebenen umgesetzt. Als Beispiel sind in der Abbildung die Auswahlkriterien *changed* und *relevance* zu sehen.

3 Gestaltung des Auswahlprozesses

Zeitpunkt der Auswahl Neben der Bestimmung des Zeitpunktes des eigentlichen Auswahlprozesses ist auch eine geeignete Repräsentation der Auswahldaten durch den gesamten Generierungsprozeß hindurch zu erarbeiten. Dieser Auswahlprozeß, d.h. die Bestimmung einer Teilmenge von Testfällen oder die Priorisierung von Testfällen, kann im Prinzip an mehreren verschiedenen Stellen geleistet werden. In Abhängigkeit der gewählten Position unterscheidet sich auch die Vorgehensweise bei der Auswahl. Dabei existieren mehrere mögliche Zeitpunkte, um eine Auswahl bzw. Priorisierung von Testfällen durchzuführen. Zunächst einmal wurde ja das Systemmodell um die schon

besprochenen Annotationen und die dazugehörigen Use-Case Diagramme erweitert. Es wurden nun drei in Frage kommende Zeitpunkte identifiziert: Vor, während und nach der Generierung der Testfälle. Im ersten Fall beschneidet man den, aus dem UML-Modell generierten, Graphen um bestimmte Zweige, was zur Folge hat, daß der weitere Prozeß der Testfallgenerierung nur noch in einem eingeschränkten Suchraum operieren muß und zusätzlich die Auswahlinformationen nicht durch den weiteren Generierungslauf getragen werden müssen. Jedoch stellt sich hier die Frage, inwieweit eine beliebig granulare Auswahl stattfinden kann, da die einzelnen Testfälle zu diesem Zeitpunkt noch nicht vorliegen. Als weitere Möglichkeit bietet sich an, die Auswahl am Ende der Tool-Chain zu leisten, wo die einzelnen Testfälle bereits in einer abstrakten XML-Repräsentation vorliegen und in einem letzten Schritt nur noch in die Zielsprache überführt werden müssen. Dabei bestehen an dieser Stelle nun zwei Vorgehensweisen; in der ersten werden die Angaben und Werte aus den Annotationen geeignet hinterlegt, um dann anschließend – basierend darauf – eine entsprechende Auswahl treffen zu können. In einem zweiten Ansatz werden nicht die Auswahlinformationen repräsentiert, sondern es wird versucht, diejenigen Stationen und Elemente aus den Diagrammen, die ein Test durchläuft, in einer geeigneten Weise zu extrahieren. D.h. man definiert eine Art der Darstellung mit deren Hilfe man beschreibt, „was“ die Fälle im einzelnen abarbeiten. Der Vorteil dieses Verfahrens ist: Werden die Prioritäten bzw. Kriterien für den Auswahlprozeß geändert, muß keine neue Generierung der Testfälle mehr erfolgen. Die Generierung ist somit komplett unabhängig von dem Auswahlverfahren, jedoch wird dafür ein erhöhter Mehraufwand bei der Datenhaltung und –Behandlung betrieben. In einem letzten Ansatz zur Auswahl soll kurz die Möglichkeit angesprochen werden, die Testfallauswahl und die Generierung eng mit einander zu verzahnen, weil schon bei der Generierung eine gewisse Auswahl statt findet. Dies bringt einen Performancevorteil (Speicherbedarf, Rechenzeit etc.) mit sich, allerdings ist dadurch eine unabhängige Implementierung beider Module nicht möglich.

Wir haben uns für eine Auswahl am Ende des Generierungsprozesses entschieden, weil dann zum Einen die Auswahl von der Generierung entkoppelt ist und zum Anderen eine erneute Generierung der Testfälle bei einer Änderung der Auswahlinformationen in den UML-Modellen nicht mehr statzufinden hat.

Zugriffsverfahren Zur Beschreibung des Inhalts der Testfälle haben wir die generierten IDs in dem erzeugten XMI-File identifiziert. Jede ID repräsentiert jedes Element eindeutig, was einen Zugriff auf das Element aus der UML-Darstellung und wahlweise somit auch auf den Inhalt der dazugehörigen Annotationen ermöglicht. Somit werden nun alle Elemente, die ein Testfall durchläuft, mit Hilfe von IDs diesem zugeordnet und in der abstrakten Repräsentation mit hinterlegt. Die Gesamtheit der IDs eines Testfalls beschreibt bzw. beinhaltet die Semantik, mit deren Hilfe nun eine Auswahl laut den hinterlegten Kriterien leistbar ist, und stellt damit auch die Verlinkung zu den Annotationen dar. Über jeweilige Zugriffsfunktionen auf die abstrakte Repräsentation einerseits und auf die XMI-Export-Datei aus dem UML-Tool können die zugehörigen Auswahlinformationen aus den Annotationen den jeweiligen Testfällen nun zugeordnet werden. Mit entsprechender Gewichtung und Bewertung der Inhalte kann nun eine Auswahl bzw. Priorisierung geleistet werden. Die Erprobung dieses Verfahrens soll demnächst in unserem Demonstrator gezeigt werden.

Verbindung zu existierenden Auswahlalgorithmen Die eigentliche Auswahl muß jetzt durch hinreichend geeignete Algorithmen gesteuert werden. In einem ersten Schritt wollen wir einen durch den Testingenieur beeinflussbaren Auswahlprozeß, indem dieser die Kriterien frei (d.h. mit Hilfe seiner gemachten Erfahrungen und den aktuellen Anforderungen) gewichten kann, implementieren. In einem nächsten soll dies durch intelligente bzw. auf bekannten risikobasierten Verfahren stützende Algorithmen ergänzt werden. Die Informationen zur Errechnung der Risiken bzw. Kosten kommen dann einerseits aus möglichen weiteren - manuell hinterlegten - Angaben in den Annotationen bzw. lassen sich andererseits aus dem Modell selbst ableiten. D.h. durch beispielsweise die Komplexität oder Größe von Funktionen aus dem Modell kann die Relevanz für einen Testlauf abgeleitet werden; komplexe Funktionen oder oft aufgerufene Funktionen sind entsprechend zu priorisieren. Für die Ermittlung und Verarbeitung dieser modellinhärenten Informationen müssen die geeigneten Verfahren noch implementiert werden.

4 Future Work

Als nächstes soll die komplette Implementierung und anschließende Evaluierung unseres besprochenen Ansatzes für die Testfallauswahl in einen Demonstrator erfolgen. In diesen sind bereits wesentliche Teile des Generierungsprozesses implementiert. In einem weiteren Schritt ist dann die modellinhärente Berechnung der Komplexität der Funktionen zu leisten, als auch mögliche weitere Auswahlkriterien zu identifizieren und in das Modell zu integrieren. Die Rückkopplung von Erfahrungswerten auf noch zu gruppierende Testfälle aus den Testläufen in den Auswahlprozeß würde zukünftig eine noch „maßgeschneiderte“ Auswahl von Testfällen ermöglichen.

An dieser Stelle möchten wir uns bei unserem Fördergeber, den Freistaat Bayern, der diese Arbeiten im Rahmen eines geförderten Projektes zur modellgestützten Testfallgenerierung ermöglichte, herzlich bedanken.

Literaturverzeichnis

- [Am05] Stale Amland, *Risk-based testing: Risk analysis fundamentals and metrics for software testing including a financial application case study*, Journal of Systems and Software, Vol. 53, 2000, pp. 265-273
- [CPS02] Yanping Chen, Robert L. Probert, D. Paul Sims, *Specification-based Regression Test Selection with Risk Analysis*, in the Proceeding of CASCON'02, September 2002
- [Kr07] M. Krüger, Y. Lei, M. Hudler, A. Kraas, P. Aschenbrenner, *Automated test case generation from UML system models*, submitted for publication for SOQUA 2007
- [Om05] Object Management Group (OMG), *Unified Modeling Language: Superstructure*, Version 2.0, formal/05-07-04, August 2005
- [Ot99] Dr. Ingrid B. Ottevanger, *A Risk-Based Test Strategy*, IQIIP Informatica B.V. Diemen, The Netherlands, Presented at STARWest 1999
- [Re04] Felix Redmill, *Theory and practise of risk-based testing*, Wiley InterScience, November 2004