Modellbasierte Generierung von Testfallszenarien für den aufwandsoptimierten Integrationstest

Sacha Reis[*], Andreas Metzger[*], Klaus Pohl[*,+]

[*]Software Systems Engineering
Universität Duisburg-Essen
Schützenbahn 70, 45117 Essen
{reis|metzger|pohl}@sse.uni-due.de

[+]Lero (The Irish Software Engineering Research Center)
University of Limerick, Ireland
pohl@lero.ie

1 Motivation

In der Praxis stellt der Integrationstest oft die aufwändigste Teststufe dar [Ts01]. Komponenten, die nicht Bestandteil des zu testenden integrierten Teilsystems sind, müssen teilweise durch Platzhalter und Treiber simuliert werden [Li02]. Diese bestimmen maßgeblich den Testaufwand und somit die Kosten des Integrationstests, da sie spezifiziert, entworfen und implementiert werden müssen. Die Anzahl der durchzuführenden Tests beeinflusst ebenfalls den Testaufwand. Neben den genannten Kriterien sind weitere Kriterien für den Testaufwand in Abhängigkeit der entwickelten Systeme oder anderer Entwicklungsparadigmen denkbar (z.B. Variabilität bei Software-Produktlinien). Beim Integrationstest existiert daher mehr als ein Kriterium, das bei der aufwandsoptimierten Ableitung der Testfälle berücksichtigt werden muss. Im Gegensatz zu Ansätzen, die sich mit der Erstellung neuer Integrationsstrategien beschäftigen, existieren nur wenige Ansätze zur Ableitung von Integrationstestfällen (z.B. [BB00], [Ts01], [HIM00]). Alle diese Ansätze berücksichtigen jedoch nicht die Selektion einer optimalen Menge von Integrationstestfällen bezüglich des Testaufwands.

In diesem Beitrag geben wir einen Überblick über unsere Integrationstesttechnik *ScenTED-IT*, die dem Tester ermöglicht, basierend auf einem existierenden Testmodell und unter Berücksichtigung mehrerer anpassbarer Optimierungskriterien Testfallszenarien für den aufwandsoptimierten Integrationstest automatisch zu generieren. Die Testfallszenarien repräsentieren eine Vorstufe von Testfällen und beinhalten noch keine konkreten Testdaten.

2 Die Technik *ScenTED-IT*

Unsere Technik zur Generierung von Integrationstestfallszenarien verwendet UML-Aktivitätsdiagramme als Testmodell. Das Testmodell stellt den Kontrollfluss sowie eine Zuordnung der darin enthaltenen Aktivitäten zu den Komponenten des Systems dar. Die Technik besteht aus fünf Aktivitäten (siehe Abbildung 1). Zunächst werden mit Hilfe des Testmodells und der Information über die Komponenten des integrierten Teilsystems alle ausführbaren Pfade durch das Testmodell bestimmt, die Interaktionen zwischen den

integrierten Komponenten beinhalten. In Abhängigkeit von der Größe des Testmodells werden entweder alle Pfadkombinationen ermittelt, die eine Abdeckung aller Interaktionen zwischen den integrierten Komponenten gewährleisten, oder aber es wird eine mathematische Optimierung mit Hilfe weniger ausgewählter Optimierungskriterien durchgeführt (siehe z.B. [WHL89]). Bei der Ermittlung aller gültigen Pfadkombinationen kann in einem weiteren Schritt mit Hilfe beliebiger Optimierungskriterien die optimale Kombination von Integrationstestfallszenarien selektiert werden.

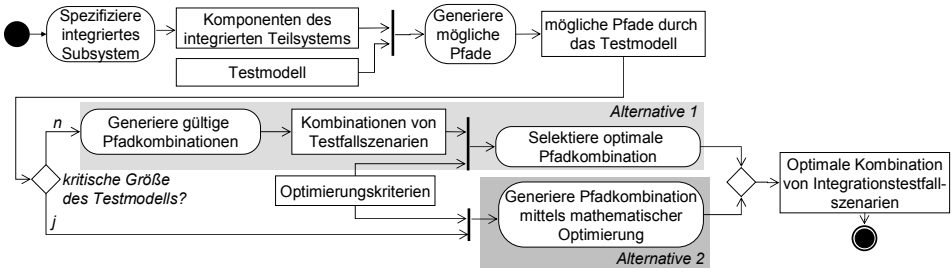

Abbildung 1: Überblick über die Technik *ScenTED-IT*

Die Vorgehensweise mit Hilfe der Generierung aller gültigen Pfadkombinationen (Alternative 1) hat den Vorteil, dass die Optimierungskriterien in Abhängigkeit des zu entwickelnden Systems und den jeweiligen Rahmenbedingungen anpassbar sind. Diese Flexibilität erfordert aufgrund der eventuell großen Menge an generierten Pfadkombinationen eine höhere Berechnungszeit, was bei einem zu großen Testmodell die Vorgehensweise der mathematischen Optimierung (Alternative 2) erfordert. Diese ist performanter, allerdings weniger flexibel bei der Wahl der Optimierungskriterien.

Die *ScenTED-IT*- Technik ermöglicht die automatisierte Ableitung von Integrationstestfallszenarien. Beide genannten Alternativen sind prototypisch implementiert worden. Bei Messungen, die unter Anwendung der Prototypen durchgeführt wurden, wurde die kritische Größe des Testmodells bestimmt. Mit Hilfe der flexibleren Alternative 1 konnten in 24 Stunden maximal Testmodelle mit 73 zu betrachtenden Pfaden berechnet werden. Wird diese Größe überschritten, muss die Alternative 2 gewählt werden.

Literaturverzeichnis

[BB00] Basanieri, F.; Bertolino, A.: A Practical approach to UML-based derivation of integration tests, In: *Proc. of the Quality Week Europe (Brussels, Belgium, November 2000)*, paper 3T, 2000.

[HIM00] Hartmann, J.; Imoberdorf, C.; Meisinger, M.: UML-Based Integration Testing, In: *Proc. of the Intl. Symposium on Software Testing and Analysis (Portland, USA, August 2000)*, pp.60-70, ACM Press, 2000.

[Li02] Liggesmeyer, P.: *Software-Qualität – Testen, Analysieren und Verifizieren von Software*. Spektrum Akademischer Verlag, 2002.

[Ts01] Tsai, W.T.; Bai, X.; Paul, R.; Shao, W.; Agarwal, V.: End-To-End Integration Testing Design. In: *Proc. of the 25th Intl. Computer Software and Applications Conf. (Chicago, USA, October 2001)*, IEEE Computer Society, pp. 166-171, 2001.

[WHL89] Wang, H.S.; Hsu, S.R.; Lin, J.C.: A Generalized Optimal Path-Selection Model for Structural Program Testing. *Journal of Systems and Software*, Vol. 10, pp.55-63, 1989.