

Dissertationen

Sami Beydeda (*sb@stecc.de*):

The Self-Testing COTS Components (STECC) Method

Promotion: Universität Leipzig, Fakultät für Mathematik und Informatik, Institut für Informatik, Lehrstuhl für Angewandte Telematik / e-Business, <http://www.lpz-ebusiness.de>

Erstgutachter: Prof. Dr. Volker Gruhn, Universität Leipzig

Zweitgutachter: Prof. Dr.-Ing. Wilhelm G. Spruth, Universität Leipzig

Drittgutachter: Prof. Dr.-Ing. Fevzi Belli, Universität Paderborn

Datum der Prüfung: 4. Dez. 2003

Veröffentlichung: Verlag Martin Meidenbauer, ISBN 389975462X

Kurzfassung:

Software-Komponenten, insbesondere kommerzielle, so genannte Commercial-Off-The-Shelf (COTS) Komponenten, stellen besondere Anforderungen an Methoden und Techniken des Testens. Einer der Gründe liegt im Informationsfluss zwischen dem Entwickler einer Komponente und ihrem Nutzer, der die Komponenten bei der Entwicklung eines größeren Systems eingesetzt, begründet. Häufig werden nicht alle notwendigen Informationen offen gelegt zu Zwecken wie Schutz des geistigen Eigentums. Unzureichender Informationsfluss kann zu vielfältigen Problemen führen, die es erforderlich machen können, Komponenten vor ihrer Integration in ein Software-System zu testen. Er kann aber nicht nur Tests durch den Nutzer der Komponente erforderlich machen, sondern auch das Testen erschweren. Ein offensichtliches Beispiel ist das Fehlen des Source Codes der Komponente für das implementationsorientierte Testen.

Die *Self-Testing COTS Components (STECC) Methode* setzt an dieser Stelle an. Der Komponenten-Nutzer kann mit Hilfe der STECC-Methode eine Komponente nach

beliebigen Kontrollflusskriterien und damit nach eigenen Zielen und Anforderungen testen, ohne dass deren Entwickler bestimmte Informationen, wie eben den Source Code, offen legen muss. Sie erfüllt somit die Anforderungen beider Parteien, wobei der Gewinn der STECC-Methode nicht auf den Komponenten-Nutzer begrenzt ist. Sie bietet außerdem dem Komponenten-Entwickler die Möglichkeit, die Komponente genau durch die Eigenschaft der Selbsttestbarkeit von anderen zu differenzieren, was in einem kompetitiven Umfeld entscheidenden Wettbewerbsvorteil bringen kann. Die zugrunde liegende Strategie der STECC-Methode liegt in dem Erweitern einer Komponente, so dass sie neben der Geschäftslogik auch Testfunktionalität implementiert.

Die STECC-Methode ermöglicht das Testen einer entsprechend erweiterten Komponente nach beliebigen Kontrollflusskriterien. Die Testfunktionalität einer Komponente umfasst daher insbesondere die Generierung von Testfällen, die im Kontext der STECC-Methode nach dem *Binary Search-based Test Case Generation (BINTEST) Algorithmus* durchgeführt wird. Der BINTEST-Algorithmus setzt auf der Binären Suche auf und erlaubt dadurch eine sehr effiziente Generierung von Testfällen. Neben dem BINTEST-Algorithmus implementiert eine STECC-Komponente noch zusätzliche Analysealgorithmen, um beispielsweise zur Laufzeit Kontrollflussgraphen generieren oder die nach den Kriterien des Komponenten-Nutzer zu überdeckenden Pfade identifizieren zu können. Das Erweitern der Komponente wird von Seiten des Komponenten-Entwicklers durchgeführt, wobei die dafür notwendigen Vorkehrungen und technischen Voraussetzungen durch das STECC-Framework zur Verfügung gestellt werden.