

Bericht zum 19. ACM-SAC Symposium “Applied Computing”,

Nicosia, Zypern, März 2004

Track “Software Engineering - Applications, Practices and Tools”

Stefan Gruner, SIHE Swansea, stefan.gruner@sihe.ac.uk

Zum zweiten mal in Folge wurde der Softwaretechnik-Track zum ACM-SAC Symposium “Applied Computing” [1] von I.K. El-Far (Microsoft Corp.), C.-C. Chiang (Univ. Arkansas), M. Montigel (Systransis AG), H.H. Thompson (Security Innovation Corp.) und S. Gruner (SIHE Swansea) organisiert.

Aus einer Anzahl vielversprechender Einreichungen wurden nach anonymer Begutachtung (im Doppel-Blind-Verfahren, je drei Gutachter pro Einreichung) neun Beiträge ausgewählt, welche im folgenden kurz zusammengefasst werden. Die Akzeptanzrate war mit knapp 25 Prozent vergleichsweise niedrig, entsprechend dem hohen Qualitätsniveau des gesamten Symposiums.

- “Missing Requirements and Relationship Discovery through Proxy Viewpoints Model”, von Seok Won Lee und David Rine. Dieser Beitrag präsentiert einen werkzeuggestützten Ansatz zur Anforderungskategorisierung, welche es dem Softwareentwickler erlaubt, Fehler im Softwarezyklus frühzeitig zu erkennen, und zwar noch vor der Phase der Systemspezifikation.
- “A Framework to Simulate UML Models: Moving from a Semi-Formal to a Formal Environment”, von Alessandra Cavarra, Elvina Riccobene und Patrizia Scandurra. Dies ist ein Beitrag zu den bereits einige Jahre andauernden Bemühungen, die stellenweise zu vage definierte Modellierungssprachenfamilie UML zu präzisieren und somit in ihrer Nutzbarkeit zu verbessern. Die Autorinnen präsentieren ein Werkzeug zur automatischen Transformation einer geeigneten Teilsprache von UML in ausführbare abstrakte Automaten, was (in theoretischer Hinsicht) eine Semantik definiert und (in praktischer

Hinsicht) die industrielle Entwicklungsarbeit mit UML veranschaulichend erleichtern soll.

- “A Portable Virtual Machine for Program Debugging and Directing”, von Camil Demetrescu und Irene Finocchi. Dieser Beitrag zum Thema Software-Tests beschreibt, wie Test- und Debug-Aktivitäten auf die höhere Stufe einer virtuellen Maschine gehoben werden können. Im Vergleich zum herkömmlichen Testverfahren bieten Tests auf einer virtuellen Maschine vielfältigere Code-Untersuchungsmöglichkeiten und liefern dem Softwareentwickler reichhaltigere, informativere Testresultate.
- “Translating the Object Constraint Language into the JAVA Modelling Language”, von Ali Hamie. Auch dieser Beitrag befasst sich mit der Utilisierung von UML. Der Autor zeigt, wie Ausdrücke der zur UML-Familie gehörenden logischen Beschreibungssprache OCL in JAVA-Code transformiert werden können. Die so gewonnenen JAVA-Sentenzen beschreiben üblicherweise Invarianten sowie Vor- und Nachbedingungen von Methodenaufrufen und dienen dann zur Programmausführung als “Wächter” der spezifizierten Integritätsbedingungen.
- “A Relational Approach to Software Metrics”, von Marco Scatto, Alberto Silitti, Giancarlo Succi und Tullio Vernazza. Die allgemein nützliche Technik der Abstraktion und des “Hievens auf höhere Ebene” begegnet uns auch in diesem Beitrag, und zwar im Kontext von Softwaremetriken. Bis dato konnte noch keinerlei Einigung über “die richtige” Softwaremetrik erzielt werden. Anstatt die Kakophonie der “richti-

gen Metriken" um eine weitere laute Stimme zu vermehren, präsentieren die Autoren eine Internet-Applikation welche alle gängigen Metriktypen umfasst, sodass ein zu untersuchendes Softwaresystem nun mit einem einzigen Meta-Werkzeug aus verschiedenen Perspektiven ausgemessen werden kann. Da das Messwerkzeugsystem an eine Datenbank gebunden ist, lassen sich Messungen unter verschiedenen Metriken auch kombinieren und hinsichtlich ihrer Korrelationen genauer analysieren ("Data-Mining").

- "A New Approach to the BDI Agent-Based Modelling", von Chang-Hyun Jo, Guobin Chen und James Choi. Die Entwicklung von Software nach dem neueren Agentenparadigma unterscheidet sich beachtlich von der Weise nach welcher traditionelle Software entwickelt wird. Die Autoren stellen ein CASE-Werkzeug vor, welches Entwurf und Entwicklung von Agentensystemen erleichtert.
- "A Control Theory Based Framework for Dynamic Adaptable Systems", von Joao Cangussu, Kendra Cooper und Changcheng Li. Plattformunabhängigkeit und Portabilität sind heutzutage wichtige Kriterien in der Softwareentwicklung. Folgt man diesen Bestrebungen, sieht man sich oftmals mit der Schwierigkeit konfrontiert, dass verschiedene Plattformen verschiedene Performanzbedingungen an die zu installierende Software stellen, insbesondere hinsichtlich des verfügbaren Speicherplatzes. Die Autoren stellen ein Verfahren vor, mit welchem sich, zur Vermeidung von Speicherfehlern, der voraussichtliche Speicherplatzbedarf eines Programmes dynamisch (zur Laufzeit) abschätzend vorhersagen lässt. In solchen Fällen lässt sich die Programmausführung oftmals rechtzeitig auf weniger kritische, alternative Pfade umleiten, wodurch die Rate der hardwarebedingten Laufzeitfehler einer Softwareapplikation deutlich gesenkt werden kann.
- "Aspect Oriented Programming for a Component-Based Real-Life Application: A Case Study", von Odysseas Papapetrou und George Papadopoulos. Mit Hilfe einer passenden industriellen Fallstudie argumentieren die Autoren für die Hypothese,

dass das Paradigma der aspektorientierten Programmierung (AOP) das Potential hat, die traditionelle objektorientierte Programmierung (OOP) von ihrer paradigmatischen Position in der industriellen Softwareentwicklung abzulösen. Die Fallstudie bezieht sich auf die konkurrierende Entwicklung einer Datenbankapplikation und vergleicht die entsprechenden Entwicklungs-Kosten und -Resultate unter Anwendung von AOP beziehungsweise OOP.

- "Modelling Socio-Technical Specifities using Architectural Concepts", von Michael Cebulla. Dieser Beitrag vereinigt Fuzzy-Logik und Agententechnik zur Modellierung und Simulation von Human-Aktoren (d.h. Menschen) im Kontext komplizierterer Mensch-Maschine-Systeme (zum Beispiel: Flugzeug mit Pilot). Ein solcher Ansatz ist der formalen Analyse zugänglich und ermöglicht formal korrekte Schlüsse insbesondere über sicherheitskritische Eigenschaften solcher hybrider Systeme.

Wie auch im Jahr zuvor, lebte der ACM-SAC'04 Softwaretechnik-Track vom Gedankenaustausch und regen Diskussionen. Im Detail fanden sich interessante Zusammenhänge auch zwischen solchen Beiträgen, deren Titel dies nicht auf Anhieb hätten vermuten lassen. Die Fortsetzung des Softwaretechnik-Diskurses auf dem ACM-SAC'05 Symposium (<http://www.acm.org/conferences/sac/sac2005/>) wird in diesen Tagen vorbereitet: siehe "Call For Papers" in dieser Ausgabe der Softwaretechnik-Trends.

Die Track-Organisatoren bedanken sich bei allen, die international zum Gelingen des Softwaretechnik-Tracks am ACM-SAC'04-Symposium beigetragen haben. Aus dem deutschsprachigen Verbreitungsbereich dieser Zeitschrift seien namentlich erwähnt die Professoren und Doktoren Herbert Kuchen, Horst Lichter und Manfred Münch.

Literatur:

- [1] H. Haddad, G. Papadopoulos, A. Omicini, R. Wainwright, L. Liebrock, M. Palakal, A. Andreou and C. Pattichis (Eds.), Proc. of the 2004 ACM Symposium on Applied Computing, Nicosia (CY), March 2004. ACM ISBN 1-58113-812-1. <http://www.acm.org/conferences/sac/sac2004/>