Konferenzbericht ETAPS 2001

Stefan Sauer Reiko Heckel
Universität Paderborn, Fachbereich Mathematik – Informatik, D 33095 Paderborn
{sauer|reiko}@uni-paderborn.de

Allgemeines

Die ETAPS-Multikonferenz 2001 (Fourth European Joint Conferences on Theory and Practice of Software) fand vom 2.–6. April 2001 in Genua statt. Das abwechslungsreiche Programm lockte über 500 Teilnehmer in die Hafenstadt an der italienischen Riviera, die einige Wochen später als Ausrichtungsort eines G8-Gipfels international Furore machen sollte.

Das Programm der ETAPS 2001 umfasste 5 Teilkonferenzen, 10 Workshops und 5 Tutorials. Eine Podiumsdiskussion beschäftigte sich mit dem Thema Opensource-Software. Tutorials wurden zu den Themen *Common Framework Initiative for Algebraic Specification and Development of Software*, *Abstract State Machines*, *Rule-based Programming Using ELAN*, *Mathematical Foundations for Software Architecture* und *Rigorous Analysis and Design with the Unified Modeling Language (UML)* abgehalten.

Konferenzen

Teilkonferenzen der ETAPS 2001 waren:

- CC 2001 – International Conference on Compiler Construction
- ESOP 2001 – European Symposium On Programming
- FASE 2001 – Fundamental Approaches to Software Engineering
- FOSSACS 2001 – Foundations of Software Science and Computation Structures
- TACAS 2001 – Tools and Algorithms for the Construction and Analysis of Systems

Die Programmkommitees der fünf Teilkonferenzen hatten insgesamt über 400 Einreichungen zu begutachten und stellten hieraus ein Programm aus 130 technischen Papieren und neun Tool-Demonstrationen sowie sieben eingeladenen Vorträgen zusammen. Mit 36 Papieren und drei Demos war TACAS mit Abstand die Größte der Einzelkonferenzen.

Die CC befasst sich traditionell mit praktischen Methoden und Werkzeugen für die Implementierung von Programmiersprachen. Schwerpunkte waren dieses Jahr Programmanalyse, -architektur und -transformation sowie verbesserte Nutzung von parallelen Prozessoren und Speicherhierarchien.

Die Themen der ESOP kreisen um Entwurf, Spezifikation und Analyse von Programmiersprachen und -systemem. Wesentliche Konferenzinhalte waren sprachbasierte Sicherheit, semantische Modellierung von Sprachmerkmalen, (statische) Programmanalyse, Nebenläufigkeit und Logikprogrammierung.

Theoretische Grundlagen von Programmen und Softwaresystemen stehen im Mittelpunkt von FOSSACS. Die behandelten Themen reichten von Nebenläufigkeit und Mobilität über Automaten und Transitionssysteme, Semantik und formale Verifikation hin zu Kategorien- und Typtheorie.

TACAS fokussiert auf Werkzeuge und Algorithmen für die Spezifikation, Verifikation, Analyse und Konstruktion von Hardware- und Softwaresystemen. Die dominierenden Themenbereiche waren in diesem Jahr Model Checking und Verifikation sowie Test- und Implementierungstechniken.

Als Hauptthemenbereiche im Querschnitt der Teilkonferenzen können Model Checking, Programmanalyse und -verifikation und verteilte Systeme identifiziert werden.

Aus dem Blickwinkel der Softwaretechnik ist insbesondere das Programm der FASE interessant. Sie beschäftigt sich schwerpunktmäßig mit der Anwendung von theoretisch fundierten Techniken im praktischen Software Engineering. Herausragendes Themengebiet war in diesem Jahr die formale Präzisierung sowie mögliche Erweiterungen der Unified Modeling Language (UML). Andere Sitzungen thematisierten die komponentenbasierte Entwicklung verteilter Systeme, Testen, Metamodellierung und formale Methoden in der Softwaretechnik. Wir werfen an dieser Stelle einen detaillierteren Blick auf das Programm dieser Teilkonferenz.

Zum Themengebiet *Metamodellierung* präsentieren Paige und Ostroff ein Metamodell für die Modellierungssprache BON in der formalen Spezifikationssprache PVS. Clark, Evans und Kent stellen den MML-Kalkül als operationale Semantik für die Metamodellierungssprache MML vor. MML basiert auf einem Übersetzungsansatz und soll als Kernsprache für die präzise Spezifikation der UML 2.0 dienen.

Für die *komponentenbasierte Entwicklung* geben Pinger und Ehrich eine algorithmische Methode an, in der Korrektheitsbeweise durch den Vergleich der Kommunikationsanforderungen mit den Kommunikationsmustern der Systemspezifikation geführt werden. Reed und Sinclair entwickeln ein formales Framework für die Beschreibung und Verifikation von Beziehungen zwischen Komponentenspezifikationen. Als theoretische Basis für Untersuchungen zur Komponierbarkeit wird das Fehlermodell von CSP benutzt. Inverardi und Uchitel stellen ein Framework für die Integration autonomer Komponenten basierend auf deren Umgebungsanforderungen vor, wobei ihre Methode mit polynomialer Speicherplatzkomplexität auskommt.

Unter den Beiträgen zur *Unified Modeling Language* präsentieren Eshuis und Wieringa eine formale Echtzeitsemantik für UML-Aktivitätendiagramme für die Modellierung von Geschäftsprozessen basierend auf einer operationalen Semantik für UML-Statecharts. Fischer, Olderog und Wehrheim ergänzen die formale Methode CSP-OZ zur Spezifikation von Komponenten um eine formal integrierte, graphische Architekturspezifikation, indem sie eine Übersetzung von UML-RT-Strukturdiagrammen nach CSP angeben. Diese basiert auf einer Formalisierung der UML-RT-Syntax in Object-Z. Heckel und Sauer beschreiben in ihrem Beitrag eine Formalisierung von UML-Kollaborationsdiagrammen mit Graphtransformationssystemen und Graphprozessen, die eine integrierte Semantik für die modellierten Strukturen, Interaktionen und Zustandstransformationen (in Form von Vor- und Nachbedingungen) angibt. Eine semiformale Methode zur Entwicklung von Komponentensystemen basierend auf dem KORRIGAN-Formalismus wird von Choppy, Poizat und Royer vorgestellt. Hierarchisch strukturierte Views für interne (statische und dynamische) und externe Aspekte von Komponenten werden in Kompositionsdiagrammen, die an UML angelehnt sind, modelliert. Perdita Stevens diskutiert in ihrem Beitrag die präzise Definition von Use Cases und deren Beziehungen in der UML und stellt einen Ansatz für deren Formalisierung vor. Bordeleau und Corriveau schlagen ein dreistufiges Vorgehen zum Übergang von szenarienorientierten Modellen des Systemverhaltens zu Verhaltensspezifikationen der interagierenden Komponenten vor. Es basiert auf der Identifizierung von Beziehungstypen zwischen Szenarien, die in verschiedene Strukturen von hierarchischen Zustandsautomaten übersetzt werden. Im Beitrag von Reggio, Cerioli und Astesiano wird ein Schema für die Semantikdefinition von UML angegeben, bei dem die individuelle Semantik verschiedener Diagrammtypen kombiniert wird. Exemplarisch wird die algebraische Spezifikationssprache CASL für die semantische Integration von Klassen- und Statechart-Diagrammen eingesetzt. Jan Jürjens verwendet die UML-Erweiterung UMLsec zur Beschreibung von Sicherheitsanforderungen, indem er Konzepte aus formalen Methoden für Sicherheitsebenen und Sicherheitsprotokolle mit den Erweiterungsmechanismen der UML umsetzt.

Ralf Lämmel untersucht das *Testen* von Grammatiken anhand von Überdeckungsanalyse und Generierung von Testmengen sowie die Integration von Testen und Grammatiktransformationen basierend auf dem Konzept von kontextabhängiger Pfadüberdeckung. Loginov et al. stellen ein Tool für die Laufzeitüberprüfung von C-Programmen vor, das den Laufzeittyp von gespeicherten Daten überwacht, prüft, aktualisiert und mit dem statischen Typ vergleicht. Niese et al. präsentieren einen Ansatz für automatisiertes, integriertes Testen von Systemen und dessen Realisierung in der integrierten Testumgebung ITE. Der Ansatz kombiniert die Paradigmen bibliothekbasierter Testentwurf, inkrementelle Formalisierung und bibliothekbasierte Konsistenzprüfung (mittels Model Checking).

Unter der Rubrik *formale Methoden* wird in dem Papier von Schröder et al. die Semantik von Architekturspezifikationen in CASL und deren Ergänzung um eine erweiterte statische Analyse thematisiert. Mossakowski, Autexier und Hutter erweitern Development-Graphen um eine Hiding-Operation. Sie entwickeln eine Semantik und ein regelgestütztes Beweissystem, welche die Dekomposition der betroffenen Spezifikationen unterstützen. Jacobs und Poll beschreiben in ihrem Beitrag eine erweiterte Hoare-Logik, um Spezifikationen in der Java Modeling Language (JML), einer IDL für Java, zu verifizieren. Mit der JML annotierte Java-Klassen werden in logische Theorien eines Theorembeweisers (z.B. PVS oder Isabelle) übersetzt. Reus, Wirsing und Hennicker stellen einen neuen Hoare-Kalkül vor, in dem Java-Klassen Zusicherungen in der Object Constraint Language (OCL) erhalten. Eine Realisierungsbeziehung zwischen Java- und Design-Subsystem wird definiert, deren Korrektheit mittels Verifikation von Trace-Abhängigkeiten, Vor- und Nachbedingungen und OCL-Invarianten gezeigt wird.

Zwei *Fallstudien* runden das Programm der FASE 2001 ab: Sharygina, Browne und Kurshan geben Entwurfsregeln für OOA-Modelle mit ausführbarer Semantik an, aus denen Automatenmodelle abgeleitet werden, die ein Model Checking erlauben. Ölveczky et al. benutzen das *Real-Time Maude*-Tool und die formale *Maude*-Methode für Spezifikation und Analyse von Multicast-Protokollkomponenten der AER/NCA Protokoll-Suite für aktive Netzwerke.

Workshops

Neben den Teilkonferenzen gehörten zehn Workshops zum Programm der Konferenz. Diese beschäftigten sich mit Themen wie algebraische, coalgebraische und relationale Entwicklungsmethoden, Spezifikationstechniken für Graphprozesse, Verifikation für Mobilität, Integrationsplattform für Werkzeuge, Java-Optimierung für eingebettete Systeme, Modellierung und Analyse von agentenbasierten Systemen, Sprachenentwicklung und Transformationen in UML.

Eingeladene Vorträge

Jede der Teilkonferenzen hatte einen eingeladenen Vortrag im Programm: Ole Lehrmann Madsen (CC): *Virtual Classes and their Implementation*, John Mitchell (ESOP): *Probabilistic Polynomial-Time Process Calculus for Security Protocol Analysis*, Bran Selic (FASE): *Physical Programming: Beyond Mere Logic*, Gordon Plotkin (FOSSACS): *Adequacy for Algebraic Effects* und Michael Fourman (TACAS): *Propositional Reasoning*. Zwei weitere Sprecher wurden für tagungsübergreifende Vorträge eingeladen. Es handelte sich hierbei um Luca Cardelli von Microsoft Research und Moshe Y. Vardi von der Rice University. Den im Titel versprochenen „Final Showdown“ zum Thema *Branching vs. Linear Time* blieb *Vardi* in seinem Vortrag schuldig. Diese Beobachtung wurde durch eine kontroverse Diskussion über die „ultimative“ Sprache zur temporalen Spezifikation und für das Model Checking einer solchen Spezifikation im Anschluss an die Präsentation belegt.

Interessanter aus softwaretechnischer Sicht war der Beitrag von *Luca Cardelli*, der in seinem Vortrag über *Globalität* drei wesentliche Bereiche identifizierte: globale Kommunikation, Global Computing und globale Daten.

Globale Kommunikationsnetzwerke (WAN, mobile Netzwerke) funktionieren nicht immer korrekt, auch wenn sich alle beteiligten Einheiten wie gefordert verhalten. Die in mobilen Netzwerken plötzlich auftretende Unmöglichkeit zur Kommunikation ist eine typische Betriebssituation. Sie erfordert eine Erweiterung des Fehlermodells dahingehend, dass sich Einheiten in einer falschen Umgebung befinden.

Die Kernfrage für *Global Computing* besteht darin, wie die Eigenschaften und Beschränkungen globaler Kommunikation in ein Berechnungsmodell eingebettet werden können. Cardelli schlägt als Lösung den Ambient-Kalkül vor. Die Ambient-Logik ist eine modale Raum-Zeit-Logik zur Beschreibung struktureller und ablaufbezogener Eigenschaften in verteilten und mobilen Systemen. Räumliche Strukturen werden als geschachtelte Regionen repräsentiert. Dann manifestiert sich Mobilität als die Veränderung von räumlichen Strukturen über die Zeit. Kommunikation und Sicherheitsaspekte werden relativ zu diesen Strukturen definiert.

Globale Daten sind semistrukturierte Daten, z.B. hierarchisch strukturierte XML-Daten, für die flexible Typtheorien benötigt werden. Cardelli identifiziert eine Ähnlichkeit von semistrukturierten Daten und Mobile Computing: Beide benötigen selbstbeschreibende Strukturen, die als kantenmarkierte Bäume oder Graphen repräsentiert werden können.

Beide Ansätze entstammen dem WWW, das zu dynamisch ist für herkömmliche Konzepte von Daten und Computing. Als ein uniformes

Modell für Daten und Abläufe im WWW bietet sich wiederum die Ambient-Logik an. Cardelli et al. entwickelten aus der modalen Spezifikationslogik für Mobile Computing eine Abfragesprache und Abfragelogik für semistrukturierte Daten.

Aus der offensichtlichen Ähnlichkeit von Daten- und Netzwerkstrukturen schließt Cardelli ein Feld für neue Forschungs- und Entwicklungstätigkeiten. Erfolgversprechend sind vor allem föderierte Lösungen. Die wesentliche Herausforderung sieht er in der Bereitstellung der theoretischen und praktischen Infrastruktur, um solche föderierten Lösungen zu erreichen.

Auch der Vortrag von *Bran Selic* (Rational Canada) thematisierte softwaretechnisch relevante Fragestellungen. Er befasste sich mit der *Berücksichtigung physischer Aspekte beim Softwareentwurf*, vor allem bei der Entwicklung verteilter Systeme. Die reale Abhängigkeit von der technologischen Plattform und die quantitativen Eigenschaften der Systemumgebung sollten beim Softwareentwurf beachtet werden, wie dies beispielsweise in speziellen Bereichen wie (eingebetteten) Echtzeitsystemen schon heute der Fall ist. Als Lösungsansatz stellt Selic ein konzeptionelles Framework für die Spezifikation und Verarbeitung von Ressourcenanforderung und -bereitstellung basierend auf einem erweiterten Typkonzept vor.

Die physischen Eigenschaften der Software werden mit dem (quantitativen) Begriff der Dienstgüte (Quality of Service, QoS) spezifiziert. Physische Typen erweitern Typspezifikationen um QoS-Eigenschaften. Die vorgeschlagene Lösung besteht in der Integration einer generischen, technologieneutralen Spezifikation der benötigten bzw. erwarteten QoS-Eigenschaften der Laufzeitressourcen in das logische Modell. Angebotene QoS-Eigenschaften müssen unabhängig von der Implementierung definiert werden, um ein hohes Maß an Portabilität und Technologieunabhängigkeit zu erreichen. Die erforderlichen Umgebungseigenschaften werden als Anforderungen der logischen Modellelemente festgelegt. Die erforderlichen und angebotenen QoS-Charakteristika werden mit einer Realisierungsbeziehung zwischen Elementen des logischen und des physischen Modells in Relation gesetzt. QoS-Spezifikationen bilden so die Basis für eine formale Verifikation der Realisierungsbeziehungen. Sie können außerdem für die automatischen Synthetisierung einer Laufzeitumgebung verwendet werden.

Best Paper Awards

Im Rahmen des Konferenzbanketts in der Villa Spinola wurden drei Tagungsbeiträge mit Best Paper Awards ausgezeichnet. Den Preis der European Association for Programming Languages and Systems (EAPLS) erhielten Pottier, Skalka und Smith für ihr Papier *A Systematic Approach to Static Access Control* (ESOP). Von der European Association for Theoretical Computer Science (EATCS) wurde die Arbeit von Hasegawa und Kakutani zum Thema *Axioms for Recursion in Call-by-Value* (FOSSACS) prämiert. Die Auszeichnung der European Association of Software Science and Technology (EASST) ging an Gunter, Muscholl und Peled für ihren Beitrag *Compositional Message Sequence Charts* (TACAS).

Schlussbetrachtung

Zusammenfassed kann man die ETAPS 2001 ohne Zweifel als eine gelungene, inhaltlich facettenreiche und anspruchsvolle und vorbildlich organisierte Konferenz bewerten.

Zu guter letzt sei bemerkt, dass es dem Organisationskommittee um Egidio Astesiano gelungen ist, die Tagung mit einem interessanten und abwechslungsreichen Rahmenprogramm abzurunden, das insbesondere reichhaltige Einblicke in die kulinarischen Genüsse der genoisischen Gastronomie gewährte.

Dieser Eindruck sollte Aufforderung und Ermunterung sein, eine Teilnahme an der ETAPS 2002 in Betracht zu ziehen, die vom 6.–14. April in Grenoble stattfinden wird. Die Frist für Einreichungen von technischen Papieren, Demonstrationen und Tutorials endet am 19. Oktober 2001.

ETAPS 2001 Ressourcen

- Konferenz-Homepage (Konferenzen, Workshops, eingeladene Vorträge etc.): http://www.disi.unige.it/etaps2001/
- Proceedings online (Springer LNCS Bände 2027–2031): http://link.springer.de/link/service/series/0558/tocs2001.htm
- ETAPS2002-Homepage (Submission Deadline 19.10.2001): http://www-etaps.imag.fr/