

# Bericht von der

## 13. Working Conference on Reverse Engineering

Sven Wenzel

wenzel@informatik.uni-siegen.de

Praktische Informatik, Universität Siegen

Vom 23. bis 27. Oktober trafen sich Freunde, Anhänger und Interessierte des Reverse Engineering zu Pasta, Wein und Diskussionen in der italienischen Bergstadt Benevento. Dort wurde von der Universität Sannio als Gastgeber die 13. Arbeitskonferenz zum Thema Reverse Engineering angeboten.

### Ort, Gastgeber & Organisation

Der Konferenzort Benevento – rund 80 Kilometer östlich von Neapel – ließ sich schnell und einfach von Neapel oder Rom mit der Bahn erreichen. Benevento selbst ist eine Stadt mit rund 65.000 Einwohnern, deren Gründung ca. 3000 Jahre zurückliegt. Aufgrund ihrer günstigen Lage in den Bergen wurde sie oft umkämpft, belagert und erobert. Sie wechselte mehrmals ihren „Besitzer“ und war zwischendurch für rund 300 Jahre eine römische Militärkolonie, wovon heute noch einige interessante Gemäuer, Triumpfbogen, Amphitheater und Kirchen zu bewundern sind. Heute ist Benevento in erster Linie für seine guten Weine bekannt.

Gastgeber war die Universität Sannio, vertreten durch das Research Center on Software Technologies (RCOST). Die Konferenz sowie die angegliederten Veranstaltungen waren vorbildlich organisiert, so dass alles reibungslos ablief. Zwischen Stadtzentrum und Konferenzgebäude verkehrte ein Shuttle-Service, der die Teilnehmer bequem vom Hotel zu den Sitzungen beförderte. Lediglich der Internetzugang war durch knapp bemessene Ressourcen ein wenig eingeschränkt, so dass man während der Vorträge (eventuell beabsichtigt?!) weniger abgelenkt werden konnte. Das fehlende Netz wurde aber durch köstliche Bewirtung und sonniges Wetter in den Kaffeepausen erfolgreich kompensiert.

### Angegliederte Veranstaltungen

Angelehnt an die Konferenz wurden ein Doktorandensymposium, ein Tutorial zum Thema Fallstudien sowie vier internationale Workshops angeboten.

Der Workshop *Program Comprehension through Dynamic Analysis* (PCODA'06) beschäftigte sich mit dem Verständnis von Software. Besonders vorbildlich war

hierbei die Durchführung des Workshops. Die insgesamt acht Beiträge wurden von drei Sprechern zu jeweils einem Themenbereich zusammengefasst und kompakt vortragen. So wurde mit dem Konzept *weniger Präsentation und mehr Diskussion* ein regelrechter Workshop angeboten.

Beim Workshop *Empirical Studies in Reverse Engineering* (WESRE'06) war interessant zu sehen, dass man sich bewusst ist, in der Vergangenheit empirische Analysen im Reverse Engineering oft vernachlässigt zu haben. Entsprechend positiv wurde daher ihre zunehmende Beachtung aufgenommen.

Der neunte *Astrenet Aspect Analysis* (AAA) Workshop beschäftigte sich mit der aspektorientierten Analyse von Software. Dabei standen in diesem Jahr cross-cutting concerns im Vordergrund.

Zum ersten Mal fand in diesem Jahr ein Workshop zum Thema *Design Pattern Detection for Reverse Engineering* (DPD4RE'06) statt. Es ist erstaunlich, dass Entwurfsmuster bereits seit den frühen Neunzigern und deren Erkennung auch schon seit rund zehn Jahren umforscht werden, aber sich dieses Jahr erstmals eine internationale Gruppe zu diesem Thema zusammenfand. Von den altbekannten Techniken und Tools waren mit Ausnahme des Gastgebers Giuliano Antoniol leider keine Vertreter anwesend. Dafür gab es viele neue Ideen und einen deutlichen Trend zum erneuten Aufleben des Themas. In der Kritik standen in erster Linie die schwammige Empirie und die schlechte Prüfbarkeit bestehender Ansätze hinsichtlich nicht gefundener Patterns. Hierauf soll sich ein zweiter Workshop im nächsten Jahr konzentrieren: In Rahmen dieses Workshops soll auch ein Wettbewerb angeboten werden, um verschiedene Erkennungswerkzeuge anhand vorgegebener Testdaten miteinander zu vergleichen. Weitere Informationen bietet hierzu eine neu eingerichtete Mailingliste [sp-recovery@iro.umontreal.ca](mailto:sp-recovery@iro.umontreal.ca).

### Die Konferenz

Die Konferenz umfasste acht Sitzungen, zwei Leitvorträge und je eine Sitzung mit Industrievorträgen und Werkzeugpräsentationen.

Der erste Leitvortrag von Lionel C. Briand ging auf empirische Untersuchungen im Reverse Engineering ein und traf damit voll das bereits auf den Workshops diskutierte Thema. Er diskutierte die vielen Faktoren, die bei Empirie zu berücksichtigen sind, und erinnerte an die oft vergessene Relation zwischen Studie und realer Anwendung. Hausi A. Müller referierte im zweiten Leitvortrag über Vergangenheit und Zukunft des Reverse Engineering.

Das Thema empirische Studien wurde während der Konferenz immer wieder angesprochen und diskutiert. Zusätzlich gab es eine eigene Sitzung zu dem Thema. Hierin wurde unter anderem die als bestes Papier ausgezeichnete Arbeit von Kapser und Godfrey vorgestellt. Sie konnten dem im Reverse Engineering so oft untersuchten Klonen auch ein paar gute Seiten abgewinnen und untersuchten Situationen, in denen durchaus beabsichtigt geklont wird. Daraus wurden Klonmuster abgeleitet, die man ähnlich wie Entwurfsmuster in verschiedensten Projekten finden kann.

Die Erkennung von Klonen an sich wurde in der Sitzung über Programmanalyse erneut aufgegriffen. Hier wurde von Rainer Koschke und seinem Team ein neuer Ansatz vorgestellt, der mit Hilfe von Syntax-Suffix-Bäumen arbeitet. Der Ansatz ist darin motiviert, dass Token-basierte Verfahren zur Klonerkennung zwar sehr schnell aber teilweise unpräzise oder Syntax-verletzend sind und Syntaxbaum-gestützte Verfahren wiederum präzise aber meist uneffizient sind. Die Lösung aus Bremen verbindet die bestehenden Alternativen zu einem vielversprechenden Mittelweg. In der anschließenden Diskussion wurde die Effizienzfrage jedoch nicht als besonders praxisrelevant angesehen.

Dem ebenfalls sehr aktuellen Thema serviceorientierte Architekturen wurde eine eigene Sitzung gewidmet. Hier ging es insbesondere um die Migration bestehender Systeme in Richtung SOA. So wurde zum Beispiel von Li und Tahvildari ein Framework vorgestellt, das die Entwickler dabei unterstützt, Services in bestehenden Java Systemen zu identifizieren, in eigenständige Komponenten zu extrahieren und letztlich von der objekt-orientierten Struktur in eine serviceorientierte Architektur zu übertragen.

Selbstverständlich darf das Thema Softwareevolution auf einer Konferenz wie der WCRE nicht fehlen. Zu diesem Thema wurden gleich zwei Sitzungen angeboten. Die erste hatte ihren Schwerpunkt in der Visualisierung von Evolutionsinformationen, während die zweite auf die Datengewinnung aus Softwarerepositories einging. Fischer und Gall stellten ihren *EvoGraph* Ansatz vor, mit dem sie Codeänderungen, die sich über mehrere Module erstrecken, visualisieren und damit Abhängigkeiten zwischen den Modulen aufzeigen. Sie zeigen dabei nicht nur auf, wann Abhängigkeiten hinzugekommen oder weggefallen sind, sondern sie visualisieren auch, wann Funktionalität neu hinzugefügt oder entfernt wurde. D'Ambros und Lanza zeigten eine Visualisierungstechnik, mit der logische Abhängig-

keiten interaktiv untersucht werden können. Mit ihrem *Evolution Radar* können sie durch die einzelnen Module eines Systems navigieren. Die Abhängigkeit zwischen verschiedenen Dateien wird dabei durch den Abstand auf dem Radar visualisiert. Beyer und Hassan stellten noch eine animierte Visualisierung von Softwareveränderungen vor. Xie, Poshyvanyk und Marcus präsentierten eine dreidimensionale Veranschaulichung von CVS Repository Informationen. Alle diese Beiträge hatten viele bunte Bilder gemeinsam und regten eine Diskussion um Visualisierungstechniken an: Im Mittelpunkt stand die Frage, ob die visuelle Gestaltung der Ergebnisse notwendig sei oder ob sortierte Zahlenkolonnen oder statistische Auswertungen nicht ebenso gute Ergebnisse liefern können. Beantwortet werden konnte diese Frage trotz reger Beteiligung an der Diskussion leider nicht – die Gemeinde spaltet sich in zwei Lager. Festzuhalten bleibt jedoch, dass die Visualisierung gerade bei großen Systemen sehr hilfreich ist, da das Auge unterschiedliche Farben und Formen deutlich schneller erfassen kann als eine große Menge von Zahlen. Ein wirklicher Beleg für den Nutzen muss aber noch geliefert werden.

Ganesan, Muthig und Knodel vom Fraunhofer IESE stellten ein Verfahren vor, um CVS Daten hinsichtlich der Entwicklerstrukturen zu analysieren. Dabei wird untersucht, wann wer an welcher Komponente gearbeitet hat und eine sog. *ownership architecture* abgeleitet. In der Produktlinienentwicklung können diese Daten sinnvoll eingesetzt werden, um die Planung von personellen Ressourcen wie Experten oder Teamstrukturen zu unterstützen.

Eine Möglichkeit zum Identifizieren von Refactorings zwischen verschiedenen Programmversionen wurde von Xing und Stroulia vorgestellt. Sie untersuchen strukturelle Informationen in Javaprojekten, indem sie die Differenzen zwischen aufeinander folgenden Softwareversionen berechnen. Auch wenn der Name des eingesetzten Tools *UMLDiff* mehr vermuten lässt, ist der Vergleich für tatsächliche UML Diagramme (insbesondere für andere Diagrammtypen als Klassendiagramme) leider nicht gegeben. Auf die modell-getriebene Softwareentwicklung kann der Ansatz also nicht so einfach übertragen werden (vgl. [www.sidiff.org](http://www.sidiff.org)).

## Fazit

Neben den hier vorgestellten Beiträgen gab es noch andere interessante Arbeiten. Die Lektüre des Konferenzbands (ISBN 0-7695-2719-1, ISSN 1095-1350) lohnt sich. Weiteren kann man die WCRE als Konferenz für Aktive und Interessierte im Bereich Reverse Engineering nur empfehlen. Der Kreis von ca. 70 Teilnehmern in den Sitzungen ist groß genug, um ausreichend Heterogenität für neue Ideen, Kritik und Diskussionen zu bieten. Er ist aber auch klein genug, um ein familiäres Flair zu wahren und nicht zu anonym zu werden.