

SATToSE 2014 — Pre-proceedings

Advanced Techniques and Tools for Software Evolution

Extended Abstracts

	Wednesday 9 July 2014	Thursday 10 July 2014	Friday 11 July 2014
8:00–9:00	Opening		
9:00–10:00	Invited talk: Alexander Serebrenik	Invited talk: Tanja Vos	Invited talk: Marianne Huchard
10:00–10:30	Coffee and tea	Coffee and tea	Coffee and tea
10:30–12:00	Mining Session: #5, #4, #15, #20, #11	Analysis Session: #18, #9, #1, #7, #16, #17	Visualisation Session: #8, #2, #12
12:00–12:30			Hackathon results
12:30–13:30	Lunch	Lunch	Lunch
13:30–14:30	Transformation Session: #14, #10, #13	Tutorial: Anya Helene Bagge	Invited talk: Alfonso Pierantonio
14:30–15:00		Technology Showdown	Hackathon results
15:00–15:30	Hackathon intro		Metrics Session: #3, #6
15:30–16:00	Coffee and tea	Coffee and tea	
16:00–17:00	Invited talk: Leon Moonen	Social event	Coffee and tea
17:00–18:30	City tour		Closing
18:30–22:00	Hackathon		

Seminar Series, SATToSE 2014
L'Aquila, Italy, 9–11 July, 2014

Dipartimento di Informatica
Università degli Studi dell'Aquila
I-67100 L'Aquila, Italy

Vadim Zaytsev (Ed.)

Preface

SATToSE is the Seminar Series on Advanced Techniques & Tools for Software Evolution. The goal of SATToSE is to gather both undergraduate and graduate students to showcase their research, exchange ideas, improve their communication skills and attend technical sessions.

The 7th edition of SATToSE took place in L'Aquila (Italy) on 9–11 July 2014. Past editions of SATToSE saw presentations on software visualisation techniques, tools for co-evolving various software artefacts, their consistency management, run-time adaptability and context-awareness, as well as empirical results about software evolution. The attendees of SATToSE 2014 enjoyed 5 invited speakers, 19 technical presentations, a tutorial, technology showdown and hackathon sessions, as well as lively social events.

Many people contributed to the success of SATToSE 2014. I would like to truly acknowledge the Steering Committee that gave us the possibility to organise the 7th edition of the event in L'Aquila. I am also indebted to Vadim Zaytsev that managed to arrange a strong program, to the invited speakers and to the students themselves that contributed to make SATToSE 2014 a remarkable experience. Last but not least I would like to thank Ludovico Iovino and Romina Eramo that contributed to the local organisation.

9–11 July 2014

Davide Di Ruscio
SATToSE 2014

Organisation

SATToSE 2014 is hosted by Department of Information Engineering Computer Science and Mathematics (DISIM) of the University of L'Aquila, Italy.

General Chair	Davide Di Ruscio (University of L'Aquila, Italy)
Program Chair	Vadim Zaytsev (University of Amsterdam, The Netherlands)
Hackathon Chair	Alexander Serebrenik (Eindhoven University of Technology, The Netherlands)
Local organisation	Romina Eramo (University of L'Aquila, Italy) Ludovico Iovino (University of L'Aquila, Italy)

Steering Committee

Ralf Lämmel	(University of Koblenz-Landau)
Michael W. Godfrey	(University of Waterloo)
Marianne Huchard	(University of Montpellier 2)
Tom Mens	(University of Mons)
Oscar Nierstrasz	(University of Bern)
Coen De Roever	(Free University Brussels)
Vadim Zaytsev	(University of Amsterdam)

<http://sattose.org>

Contents

Lecture Abstracts

Alexander Serebrenik	
Human Aspects of Software Engineering	2
Leon Moonen	
Assessment and Evolution of Safety-Critical Cyber-Physical Product Families	3
Tanja Vos	
Test Automation at the User Interface Level	5
Marianne Huchard	
Relational Concept Analysis: Mining Multi-relational Datasets for Assisted Class Model Evolution	6
Alfonso Pierantonio	
Non-determinism and Bidirectional Model Transformations	7
Anya Helene Bagge	
Language, Models and Megamodels	8

Presentation Abstracts

Mining

Angela Lozano, Gabriela Arévalo, Kim Mens	
Co-Occurring Code Critics	10
Reinout Stevens, Coen De Roover	
QwalKeko, a History Querying Tool	14
Philipp Schuster, Ralf Lämmel	
Evolution and Dependencies of Hackage Packages	18
Çiğdem Aytekin, Tijs van der Storm	
Inheritance Usage in Java: A Replication Study	22
Juri Di Rocco	
Mining Metrics for Understanding Metamodel Characteristics	27

Transformations

Romeo Marinelli	
A Taxonomy of Bidirectional Model Transformation and Its Application	31
Gianni Rosa	
Managing Uncertainty in Bidirectional Transformations	35

Lecture Abstracts

Test Automation at the User Interface Level

Invited Talk Abstract

Tanja E. J. Vos

Departamento de Sistemas Informaticos y Computación
Universidad Politecnica de Valencia
Spain
`tvos@dsic.upv.es`

Testing applications at the UI level is an important yet expensive and labour-intensive activity. Several tools exist to automate UI level testing. Capture replay tools rely on the UI structure and require substantial programming skills and effort. These tools implicitly make the assumption that the UI structure remains stable during software evolution and that such structure can be used effectively to anchor the UI interactions expressed in the test cases. Consequently, when test cases are evolved, adapted, parameterized or generalized to new scenarios, the maintenance cost can get real high and the competence required from programmers can become an obstacle. Visual testing tools take advantage of image processing algorithms to simulate the operations carried out manually by testers on the UI making UI testing as simple as that carried out step by step by humans. These visual testing approaches simplify the work of testers as compared to the structural testing approaches. However, they do rely on the stability of the graphical appearance of the UI, and require substantial computational resources for image processing. Changes to the application often also involve changes to the UI, hence also threatening the visual approach. Visual clues in the UI might mislead the image recognizer of visual testing tools, which are correspondingly subject to false positives (wrong UI element identification) and false negatives (missed UI elements). In this talk we will present the tool TESTAR, whose development was initiated under the FITTEST project. TESTAR automatically generates and executes test cases based on a structure that is automatically derived from the UI through the accessibility API. Since this structure is build automatically during testing, the UI is not assumed to be fixed and tests will run even though the UI evolves, which will reduce the maintenance problem that threatens the approaches mentioned earlier. The basic functionality of the tool will be presented, together with some case studies done in industry to evaluate the approach.