SATToSE 2015: The Post-Proceedings Editorial

Haidar Osman[1], Davide Di Ruscio[2], Vadim Zaytsev[3],
Mircea Lungu[4], Anya Helene Bagge[5]

[1] University of Bern, Switzerland osman@inf.unibe.ch
[2] University of L'Aquila, Italy, davide.diruscio@univaq.it
[3] Raincode, Belgium, vadim@grammarware.net
[4] University of Groningen, The Netherlands m.f.lungu@rug.nl
[5] University of Bergen, Norway anya@ii.uib.no

Venue

SATToSE is the Seminar Series on Advanced Techniques and Tools for Software Evolution. Its previous editions have happened in Waulsort (Belgium, 2008), Côte d'Opale (France, 2009), Montpellier (France, 2010), Koblenz (Germany, 2011, 2012), Bern (Switzerland, 2013), and L'Aquila (Italy, 2014). Its 8th edition took place in Mons, Belgium from the 6^{th} till the 8^{th} of July, 2015. Each edition of SATToSE witnesses presentations on software visualisation techniques, tools for coevolving various software artefacts, their consistency management, runtime adaptability and context-awareness, as well as empirical results about software evolution.

The goal of SATToSE is to gather both undergraduate and graduate students to showcase their research, exchange ideas, improve their communication skills, attend and contribute technology showdown and hackathons.

The highlights of the programme included two invited talks (given by Anita Sarma and Martin Pinzger), two interactive tutorials (by Gregorio Robles and Massimiliano Di Penta), and hackathon (by Bogdan Vasilescu). The detailed programme, as well as the pre-proceedings drafts can be found on our website: http://sattose.org/2015.

Selection process

Each pre-proceedings submission was reviewed by at least three different peers. All submissions with a conflict of interest with one of the editors (co-authored by them or their colleagues) were handled by the other PC chair. We would like to express our gratitude to the program committee (listed in lexicographic order) who provided the reviews.

- Anya Helene Bagge
- Andrea Caracciolo
- Marianne Huchard
- Csaba Nagy
- Coen De Roover
- Davide Di Ruscio
- Alexander Serebrenik
- Bogdan Vasilescu
- Vadim Zaytsev

The call for post-proceedings contributions was communicated to all participants after the event. Only some decided to pursue the finalisation of their contribution for the post-proceedings where they might have solicited more co-authors, changed the title, and included more results. As a result, we have received 6 submissions of the extended versions of pre-proceedings abstracts.

Each submitted report has been reviewed by at least three different peers. All submissions with a conflict of interest with one of the editors (co-authored by them or their colleagues) were handled by the other editor. The emphasis was put on clear problem definitions and descriptions of advanced aspects of the techniques contemplated in the solution, as opposed to the finality of the obtained results. Thus, most submissions are intermediate reports on ongoing work or summaries of previously developed tools and papers.

Organisation

- **General Chair:** Tom Mens (University of Mons, Belgium)
- **Program Chair:** Anya Helene Bagge (University of Bergen, Norway)
- **Hackathon Chair:** Bogdan Vasilescu (UC Davis, USA)
- **Local Organization Chair:** Mathieu Goeminne (University of Mons, Belgium)
- **Steering Committee Chair:** Ralf Lämmel (Universität Koblenz-Landau, Germany)
- **Steering Committee:**
  - Coen De Roever (Free University Brussels, Belguim)
  - Davide Di Ruscio (University of L'Aquila, Italy)
  - Michael W. Godfrey (University of Waterloo, Canada)
  - Oscar Nierstrasz (University of Bern, Switzerland)
  - Vadim Zaytsev (Universiteit van Amsterdam, The Netherlands)
  - Tom Mens (University of Mons, Belgium)
  - Marianne Huchard (Université Montpellier 2, France)
  - Anthony Cleve (University of Namur, Belgium)
- **Post-proceedings Editors:**
  - Anya Helene Bagge (University of Bergen, Norway)
  - Tom Mens (University of Mons, Belgium)
  - Haidar Osman (University of Bern, Switzerland)

Contents of the volume

4. *Predicting Software Defectiveness through Network Analysis*
   We used a complex network approach to study the evolution of a large software system, Eclipse, with the aim of statistically characterizing software defectiveness along the time. We studied the software networks associated to several releases of the system, focusing our attention specifically on their community structure, modularity and clustering coefficient. We found that the maximum average defect density is related, directly or indirectly, to two different metrics: the number of detected communities inside a software network and the clustering coefficient. These two relationships both follow a power-law distribution which leads to a linear correlation between clustering coeffcient and number of communities. These results can be useful to make predictions about the evolution of software systems, especially with respect to their defectiveness.
5. *Interactive User-Oriented Views for Better Understanding Software Systems*
   Understanding software artefacts is a crucial task for people who want to participate in any software development process. However, because of the large amount of detailed and scattered information in software artefacts, understanding them is usually time-consuming and vulnerable to human errors and subjectivities. A system that aids practitioners to investigate understanding about software artefacts could reduce the vulnerabilities and speed up software development/maintenance process. Our research focuses on building a comprehensive view of software system in order for developers to achieve the two goals: (i) to save the time spending on searching and navigating on source code; and (ii) to gain better understanding about software artefacts regarding to domain-specific tasks. To achieve these goals, we propose an empirical approach in which the visualisation and the generation of high-level design and architectural views from source code and design documentations have been played central roles. The research is ongoing and could potentially be extended to different software artefacts (such as requirements, use-cases, test-cases, revision logs).
6. *BibSLEIGH: Bibliography of Software (Language) Engineering in Generated Hypertext*
   The body of research contributions is vast and full of papers. Existing projects help us navigate through it and relate authors to papers and papers to venues. In this paper we list features missing from those projects and propose a solution in the form of BibSLEIGH, a work in progress on facilitated browsing of scientific knowledge objects. Through leveraging domain focus, by actively employing automated data collection and scraping tools, and with automated annotating of the corpus, we are able to gain and provide insights into scientific communities and topics, as well as surface potential interdisciplinary opportunities.