8th International Workshop on Search-Based Software Testing (SBST 2015)

Gregory Gay*, Giuliano Antoniol†
* Department of Computer Science & Engineering
University of Minnesota, Minneapolis, MN, USA
Email: greg@greggay.com
† Département de Génie Informatique et Génie Logiciel
École Polytechnique de Montréal, Montréal, QC, Canada
Email: antoniol@ieee.org

***Abstract*—This paper is a report on the 8th International Workshop on Search-Based Software Testing at the 37th International Conference on Sofrware Engineering (ICSE). Search-Based Software Testing (SBST) is a form of Search-Based Software Engineering (SBSE) that optimizes testing through the use of computational search. SBST is used to generate test data, prioritize test cases, minimize test suites, reduce human oracle cost, verify software models, test service-orientated architectures, construct test suites for interaction testing, and validate real-time properties. The objectives of this workshop are to bring together researchers and industrial practitioners from SBST and the wider software engineering community to share experience and provide directions for future research, and to encourage the use of search techniques to combine aspects of testing with other aspects of the software engineering lifecycle.**

Three full research papers, three short papers, and three position papers will be presented in the two-day workshop. Additionally, six development groups have pitted their test generation tools against a common set of programs and benchmarks, and will present their techniques and results. This report will give the background of the workshop and detail the provisional program.

I. Background and Workshop Goals

There is a growing realization that optimization can be applied to many aspects of the software development process—a research area known as Search-Based Software Engineering (SBSE) [2]. Search-Based Software Testing—one of the largest research areas within SBSE—is the process of using search-based optimization algorithms to specifically address problems in software testing [3] . SBST has been applied to a wide variety of testing goals including structural, functional, non-functional and state-based properties. Many approaches to testing and a wide diverse range of development domains have been addressed, including exceptions, interactions, integration, mutation, regression, stress and web applications.

In all approaches to SBST, the primary concern is to define a fitness function (or set of fitness functions) that capture the test objectives. The fitness function is used to guide a search-based optimization algorithm, which searches the space of test inputs to find those that meet the test objectives. Because any test objective can, in principle, be re-cast as a fitness function, the approach is widely applicable—as demonstrated by the prior list of testing applications.

Work in SBST has developed to the point at which it is now ripe for combination with other areas of software engineering. The common "lingua franca" that makes these combinations possible is the definition of the fitness function. A fitness function is merely a form of metric [1], and metrics exist across the entire software engineering spectrum. Therefore, the central objective of this workshop is to bring together researchers and industrial practitioners from SBST and the wider software engineering community to share experience and provide directions for future research, and to encourage the use of search techniques to combine aspects of testing with other aspects of the software engineering lifecycle.

II. Workshop Format

SBST is a two-day workshop aimed at bringing testing researchers together with the broader software engineering community to discuss state-of-the-art work and set new research directions. The workshop will consist of two keynote addresses, several technical paper sessions, and two panel discussions.

A. Keynote Addresses

Myra Cohen (University of Nebraska at Lincoln) and Massimiliano Di Penta (University of Sannio) will give keynote addresses. Following time for questions, we will ask the speakers to raise three issues arising from their talk for participants to discuss in small groups. Discussion points will be transcribed onto the workshop website.

B. Technical Papers and Tool Competition

Three types of contributions were sought after, including:

- Full papers on original research—either empirical or theoretical—in SBST, practical experience of using SBST, or SBST tools.
- Short papers that describe novel techniques, ideas and positions that have yet to be fully developed; or are a discussion of the importance of a recently published SBST result by another author in setting a direction for the SBST community, and/or the potential applicability (or not) of the result in an industrial context.

- Position papers that analyze trends in SBST and raise issues of importance. Position papers are intended to seed discussion and debate at the workshop.

Each paper was reviewed by at least three PC members and evaluated according to the criteria of relevance, novelty, soundness, and ability to spark discussion.

Additionally, we have organized a tool competition where we invited researchers, students, and tool developers to design innovative new approaches to software test generation. The developers of these tools will present their techniques and results during the workshop.

C. Discussion Panels

We plan to hold two discussion panels, where a diverse group of experts in search-based software testing will provide perspectives on a common topic, leaving time for whole group discussion. The planned panels are:

1) SBST for Web and Mobile Software
2) SBST Tool Development

Panelists will be announced in the near-future on the workshop website (http://searchbasedsoftwaretesting.org).

D. Provisional Program

The provisional program is as follows:

Day 1:

8:45 – 9:00am: Introduction
9:00 – 10:30am: Keynote: Myra Cohen
10:30 – 11:00am: *Break*
11:00 – 12:30pm: Research Session 1

- Robert Feldt and Simon Poulding. *Broadening the Search in Search-Based Software Testing: It Need Not Be Evolutionary*.
- Erik M. Fredericks and Betty Cheng. *An Empirical Analysis of Providing Assurance for Self-Adaptive Systems at Different Levels of Abstraction in the Face of Uncertainty*.
- Yue Jia. *Hyperheuristic Search for SBST*.
- Justyna Petke. *Constraints: the Future of Combinatorial Interaction Testing*.

12:30 – 2:00pm: *Lunch*
2:00 – 3:30pm: Competition Session 1

- Urko Rueda, Tanja E. J. Vos and Wishnu Prasetya. *Unit Testing Tool Competition – Round Three*
- Gordon Fraser and Andrea Arcuri. *EvoSuite at the SBST 2015 Tool Competition*
- Annibale Panichella, Fitsum Meshesha Kifetew and Paolo Tonella. *Results for EvoSuite-MOSA at the Third Unit Testing Tool Competition*

3:30 – 4:00pm: *Break*
4:00 – 5:30pm: Panel - SBST for Web and Mobile Software

Day 2:

9:00 – 10:30am: Keynote: Massimiliano Di Penta
10:30 – 11:00am: *Break*
11:00 – 12:30pm: Research Session 2

- Shujuan Jiang, Lingsai Wang and Yanmei Zhang. *Test Case Generation Based on Particle Swarm Optimization Using Orthogonal and Local Exploration*.
- Kevin Salvesen, Juan Pablo Galeotti, Florian Gross, Gordon Fraser and Andreas Zeller. *Using Dynamic Symbolic Execution to Generate Inputs in Search-Based GUI Testing*.
- Maurizio Leotta, Andrea Stocco, Filippo Ricca and Paolo Tonella. *Meta-Heuristic Generation of Robust XPath Locators for Web Testing*.
- Mohamed Boussaa, Olivier Barais, Gerson Sunye and Benoit Baudry. *A Novelty Search Approach for Automatic Test Data Generation*.

12:30 – 2:00pm: *Lunch*
2:00 – 3:30pm: Competition Session 2

- Wishnu Prasetya. *T3: Benchmarking at Third Unit Testing Tool Contest*
- Lei Ma, Cyrille Artho, Cheng Zhang, Hiroyuki Sato, Masami Hagiya, Yoshinori Tanabe and Mitsuharu Yamamoto. *GRT at the SBST 2015 Tool Competition*
- Abdelilah Sakti, Yann-Gal Guhneuc and Gilles Pesant. *Third Unit Testing Tool competition: Results for JTExpert*

3:30 – 4:00pm: *Break*
4:00 – 5:30pm: Panel - SBST Tool Development
5:30pm: Closing Remarks

References

[1] M. Harman and J. Clark. Metrics are fitness functions too. In *Proc. International Symposium on METRICS*, pages 58–69. IEEE Press, 2004.

[2] M. Harman and B. F. Jones. Search based software engineering. *Information and Software Technology*, 43(14):833–839, Dec. 2001.

[3] P. McMinn. Search-based software test data generation: A survey. *Software Testing, Verification and Reliability*, 14(2):105–156, June 2004.