Search-Based Software Testing (SBST) ICSE 2020 Workshop Proposal

José Miguel Rojas
Department of Informatics, University of Leicester
Leicester, United Kingdom

Erik M. Fredericks
Dept. of Computer Science & Engineering, Oakland University
Rochester, MI, USA

***Abstract*—There is a growing realization that optimization techniques can be applied to many aspects of the software development process: a research area known as Search-Based Software Engineering (SBSE). 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. 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, and web 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 that guides a search algorithm. A fitness function is merely a form of a metric, 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.

SBST is a one-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.

I. Thoughts

- Theme? SBST
- Award?
- Fishbowl session

II. Contact Information

José Miguel Rojas. Address: Department of Informatics, University of Leicester, University Road, LE1 7RH, Leicester, United Kingdom. Phone: +44 (0) 116 252 3828. Email: j.rojas@leicester.ac.uk.

Erik M. Fredericks (main contact person). Address: Department of Computer Science and Engineering, Oakland University, 115 Library Dr., Rochester, MI, USA, 48309. Phone: +1 (248) 370 4075 3828. Email: fredericks@oakland.edu.

III. Motivation and Objectives

Search-Based Software Engineering (SBSE) [1] is receiving increasing attention from the SE community thanks to the growing realization that optimization can be applied to many aspects of the software development process. Search-Based Software Testing—the process of using search-based optimization algorithms to specifically address problems in software testing [2]—is one of the largest research areas within SBSE. The potential of SBST to be applied to a vast number of open complex testing problems (e.g., autonomous driving functions [3]) is increasingly evident. Furthermore, the recent industrial-scale impact of the Sapienz tool [4] at Facebookshowcases the capabilities and potential of SBST and helps as motivation for the community to continue to work on SBST techniques and tools.

Some of the specific topics to be explored in the workshop include:

Multi-Objective Solutions. SBST can use many fitness functions at once, presenting the end-user with a choice of non-dominating solutions at the so-called "Pareto Front". Multi-objective optimization may be used to produce test suites that aim to satisfy multiple goals, such as generating test suites that focus on verifying properties as well as covering all of the branches of a program [5].

Search-Based Optimization and the Test Oracle. One of the most difficult challenges in software testing is the development or synthesis of the test oracle—an automated judgment of the behavior of the software [6]. Search techniques have the potential to be used in automatic oracle generation, oracle debugging, and the generation of tests engineered to surface faults tuned to specific oracles.

SBST for Web and Mobile Software. With the rapid rise of mobile and web-based software, the need has also emerged for effective testing of those systems. Search-based techniques have the potential to make a large impact in these areas, and have been used to—for example—provide security testing of web applications [7].

Human Aspects and Integration into Real Test Environments. Search-based techniques present the possibility to take into account domain knowledge or optimize tests for human usage (for example, incorporating a language model to produce test strings that aid comprehension [8]).

Enriching SBST with Machine Learning. Machine learning techniques have proven their worth across many aspects of software development. The use of machine learning to enrich heuristic optimization algorithms—and the use of search-

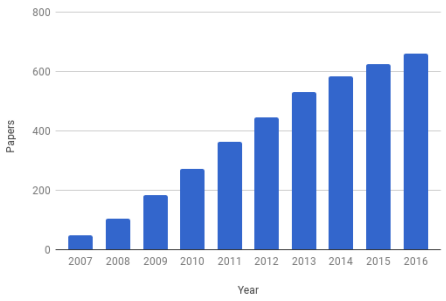

Fig. 1. Total number of publications in SBST since 2007.

based techniques to enhance learning algorithms—may be of incredible benefit to the SBST community [9].

Application of SBST in Industrial Practice and the Classroom. We have begun to see the adoption of SBST techniques in industrial settings. We would like to explore where SBST can and should be making an impact. Similarly, we would like to explore the practicality of teaching SBST as part of the software engineering or computer science curriculum.

Synergies of SBST and other SE areas. SBST is also ripe for combination with other areas of software engineering, e.g., requirements optimization and project planning. Possibilities exist to optimize the selection of requirements based not only on traditional aspects of SBSE for requirements (customer satisfaction, cost, etc.), but also on the implications for regression testing (code coverage, test execution time, and test effectiveness).

A. Relevance to Software Engineering

SBST has been applied to a wide variety of testing goals including structural [10], functional [11], non-functional [12] and state-based properties [13]. Many approaches to testing and a wide diverse range of development domains have been addressed, including exceptions [14], interactions [15], mutation [16], regression [17], stress [18] and web applications [7]. In 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. It is therefore highly relevant to Software Engineering. This relevance is further testified by the increasing cumulative number of publications in the area since 2007, as Figure 1 shows, providing evidence that SBST is a growing area [19].

B. Relevance To ICSE

The SBST workshop compliments ICSE well. The success of the previous five editions of SBST which were co-located with ICSE reassures us that the ICSE community is highly interested in computational search, and its applications in testing, verification, and a number of other Software Engineering tasks, and that the key to strenthening and expanding this exciting research area is to continue to hold our workshop alongside the ICSE conference. For example, SBST was the 3rd largest both in the number of submissions and registered participants among all workshops held at ICSE in 2016 and following the recent media impact of a SBST tool (Sapienz), we expect a growing interest amongst ICSE participants in attending our workshop. By continuing to hold SBST with ICSE, we can reach out to other areas of software engineering expertise and produce fruitful collaboration.

C. Balance and Synergy Across ICSE Events

SBST complements workshops regularly appearing at ICSE, including the Automated Software Test (AST) workshop. We believe this is for two reasons:

1) SBST is More Than Just Automation. One of the benefits of the SBST approach is the ability to include a human-in-the-loop and their domain expertise. This has been exploited in search-based test data generation, where manually-produced test cases have been used to "seed" the search. Other techniques have used resources produced from man-made artifacts, including natural language models [8]. Search-based approaches allow for humans to supply judgements and act as a fitness function in their own right.

2) Search-Based Optimization Techniques are Specialized. Optimization is a research field in itself, and the application of techniques from this domain requires specific attention. The incorporation of an optimization technique involves thinking about the problem representation and properly tuning the search operators (i.e., crossover and mutation) to ensure proper solution convergence. Thus, the specifics of applying search-based optimization to testing are unlike the issues encountered in other areas of automated software engineering. Therefore, we believe a dedicated workshop addressing these matters is of importance and interest to ICSE participants, and as such, SBST is a useful addition to the ICSE workshops.

IV. Format and Required Services

A. Workshop Format

Some of the major goals of this workshop are to stimulate discussion, seed the generation of ideas for future research in SBST, and to encourage potential working relationships between participants. The starting point for discussions will be the following components:

Keynotes. We have shortlisted Claire Le Goues (Carnegie Mellon University) (software evolution and repair), Justyna Petke (University College London, UK) (search-based software testing), Phil McMinn (University of Sheffield, UK) (automated software / database testing), Gordon Fraser (University of Passau, Germany) (suggested topic: future of search-based unit test generation), and Robert Feldt (Chalmers University of Technology, Sweden) (search-based software testing in industry) as candidate keynote speakers.

Paper Sessions. Each full paper will be given a 20 minute slot: 15 minutes will be devoted to the actual presentation.

The final 10 minutes will be allotted for questions and discussion. We will nominate a "discussant" for each paper who will prepare questions in advance and be responsible for generating discussion points. To facilitate this, papers will be distributed in advance of the workshop and participants will be encouraged to read them in advance. Each short paper will be given a 15 minute slot: 10 minutes for presentation and 5 minutes for discussion.

Tool Competition. The SBST Tool Competition has been running since 2012 and has consistently been well received by the workshop attendees. Participants apply their test generation approaches to a set of proposed benchmarks and submit a four-page report detailing their approach and results. The 8th edition of the competition will be organised by **Xavier Devroey** (Delft University of Technology, Netherlands), **Sebastiano Panichella** (University of Zurich), and **Alessio Gambi** (Universität Passau). We believe this competition will continue to attract students and researchers whose focus is to build industrial SBST tools.

Tutorial. The tutorial at SBST has grown in popularity in the recent editions of the workshop. Following Gordon Fraser (2016), Lionel Briand (2017), Andrea Arcuri (2018), and Nadia Alshahwan (2019), we have preliminary acceptance from Matas Martnez to present his work on Astor, an automated fault repair framework, and how its techniques may be incorporated into the SBST domain..

B. Intended Length

We propose to hold a one-day workshop. Our preference is to hold the workshop before the main conference on May 23 or 24, as we did in previous editions.

The proposed length of the workshop is in line with our expectation of accepting approximately four full papers, six short/position papers, and four tool submission papers. A tentative program is therefore as follows:

Schedule:	
8:45 – 9:00am	Introduction
9:00 – 10:30am	Keynote
10:30 – 11:00am	*Break*
11:00 – 12:30pm	Paper Session 1 (4 papers)
12:30 – 2:00pm	*Lunch*
2:00 – 3:30pm	Paper session 2 (6 papers)
3:30 – 4:00pm	*Break*
4:00 – 5:30pm	Tool competition
5:30pm	Close

C. Logistics

We require a projector and projector screen for the presenters. Presenters will be able to use their own laptops, but will be able to use an organiser's, if necessary. A flipboard and/or whiteboard would be useful for recording discussion or making announcements.

V. Participant Solicitation

We intend to run an open workshop, in which papers will be solicited for presentation. We plan to make a call for full papers, short papers, and position papers and request at least one author of each accepted contribution to register and attend the workshop. Participation will not be limited to presenters—workshop attendance will be open to all. We will especially encourage participation from students in all categories. While SBSE and SBST will likely be the dominant background of the participants, we will make an effort to attract researchers from other domains to foster wider discussions.

A. Expected Participant Numbers

We expect to maintain the high level of attendance of previous editions of the workshop. As a reference, the 2018 edition of workshop, held before the ICSE 2018 main conference in Gothenburg, Sweden, attracted approximately 30 participants.

VI. Proceedings

A. Expected Contributions

We expect to accept 12–16 papers comprising of full papers (8 pages), short papers (4 pages), position papers (2 pages) and competition reports (4 pages). Accepted papers will appear in a pre-proceedings and will be proposed for publication in the ACM/IEEE digital libraries.

B. Review and Evaluation Process

We will follow a follow a standard bidding process. Each paper will be reviewed by at least three PC members and evaluated according to the criteria of relevance, novelty, soundness, and ability to spark discussion during the workshop. Following reviews, there will be an online discussion, and finally, the organizers will make final decisions on paper acceptance, based on referee reviews and conclusions of the discussions. Depending on the level of submissions, we may also invite poster presentations from borderline papers that do not meet the acceptance criteria.

C. Program Committee

We plan to invite senior and young researchers to serve on the Program Committee. To guarantee the continued success of the SBST workshop and to keep growing the community, we aim to keep many of the people who have been served as PC members in previous editions of the workshop:

- Justyna Petke, *University College London, United Kingdom*
- Gregory Gay, *University of South Carolina, United States*
- Giuliano Antoniol, *École Polytechnique Montŕeal, Canada*
- Mark Harman, *Facebook London, United Kingdom*
- Tanja Vos, *Universidad Politécnica de Valencia, Spain*
- John Clark, *University of Sheffield, United Kingdom*
- Gordon Fraser, *University of Passau, Germany*
- Erik Fredericks, *Oakland University, United States*
- Phil McMinn, *University of Sheffield, United Kingdom* - Remove?
- Paolo Tonella, *Università della Svizzera italiana, Switzerland*
- Annibale Panichella, *Delft University of Technology, Netherlands*
- Myra Cohen, *Iowa State University, United States*
- Nazareno Aguirre, *Universidad Nacional de Rio Cuarto, Argentina*
- Gabriela Ochoa, *University of Stirling, United Kingdom*
- Sebastiano Panichella, *University of Zurich, Switzerland*
- Claire Le Goues, *Carnegie Mellon University, United States*
- Thomas Vogel, *Humboldt-Universitat zu Berlin, Germany*
- Byron DeVries, *Grand Valley State University, USA*

D. Website

The website of the proposed workshop will be at:
http://www.searchbasedsoftwaretesting.org/2020 .

VII. Workshop History

From 2008 to 2013, SBST was co-located with ICST (Intl. Conference on Software Testing, Verification and Validation). Since 2014, SBST has been co-located with ICSE:

Ed.	Website
12th	https://sbst19.github.io
11th	http://software.imdea.org/sbst18/
10th	http://sbst2017.lafhis.dc.uba.ar/
9th	https://cse.sc.edu/~ggay/sbst2016/
8th	http://sbst2015.soccerlab.polymtl.ca/
7th	http://www.searchbasedsoftwaretesting.org/2014/

VIII. Proposers' Bios

José Miguel Rojas is an Assistant Professor (Lecturer) at the University of Leicester, United Kingdom. Previously, he was a Research Associate in Software Testing at The University of Sheffield, working mainly on search-based automated test generation and its application in real-world software development scenarios. José received his PhD in Software and Systems from the Technical University of Madrid (Spain, 2013). His research interests include empirical software engineering, automated software testing, and software engineering education. His work has been published in the top venues of logic programming (ICLP), software engineering (ICSE and ASE), software testing (ISSTA and ICST) and search-based software engineering (SSBSE and GECCO). He has co-chaired the MUTATION 2017 and MUTATION 2018 workshops and the SSBSE 2018 Challenge Track.

Erik Fredericks is an Assistant Professor at Oakland University in Rochester, MI, USA. He received his Ph.D. from Michigan State University in Computer Science and Engineering in 2015, focusing on search-based software engineering (including testing) and self-adaptive systems. His research interests also include cyber-physical systems, model-driven engineering, and evolutionary computation. Before joining Oakland University, he was a postdoctoral researcher at Michigan State University as well as a software engineer in the automotive industry. Erik co-chaired the Natural Language Processing for Software Engineering (NL4SE) workshop in 2018, has served as publicity/social media chair for FSE 2020 and SSBSE 2018, served as web co-chair of ICSE 2013, and regularly serves as a program committee member for several conferences.

References

[1] M. Harman and B. F. Jones, "Search based software engineering," *Information and Software Technology*, vol. 43, no. 14, pp. 833–839, Dec. 2001.

[2] P. McMinn, "Search-based software test data generation: A survey," *Software Testing, Verification and Reliability*, vol. 14, no. 2, pp. 105–156, Jun. 2004.

[3] R. B. Abdessalem, A. Panichella, S. Nejati, L. C. Briand, and T. Stifter, "Testing autonomous cars for feature interaction failures using many-objective search," in *Proceedings of the 33rd ASE*, 2018, pp. 143–154.

[4] K. Mao, M. Harman, and Y. Jia, "Sapienz: Multi-objective automated testing for android applications," in *Proceedings of the 25th International Symposium on Software Testing and Analysis*, ser. ISSTA 2016, 2016, pp. 94–105.

[5] G. Gay, M. Staats, M. W. Whalen, and M. P. E. Heimdahl, "Moving the goalposts: Coverage satisfaction is not enough," in *Proceedings of the 7th International Workshop on Search-Based Software Testing*, ser. SBST 2014. New York, NY, USA: ACM, 2014, pp. 19–22.

[6] M. Harman, P. McMinn, M. Shahbaz, and S. Yoo, "A comprehensive survey of trends in oracles for software testing," University of Sheffield, Department of Computer Science, Tech. Rep. CS-13-01, 2013.

[7] J. Thomé, A. Gorla, and A. Zeller, "Search-based security testing of web applications," in *Proceedings of the 7th International Workshop on Search-Based Software Testing*, ser. SBST 2014. New York, NY, USA: ACM, 2014, pp. 5–14.

[8] S. Afshan, P. McMinn, and M. Stevenson, "Evolving readable string test inputs using a natural language model to reduce human oracle cost," in *International Conference on Software Testing, Verification and Validation (ICST 2013)*. IEEE, March 2013. [Online]. Available: http://philmcminn.staff.shef.ac.uk/publications/pdfs/2013-icst.pdf

[9] R. Malhotra, "Search based techniques for software fault prediction: Current trends and future directions," in *Proceedings of the 7th International Workshop on Search-Based Software Testing*, ser. SBST 2014. New York, NY, USA: ACM, 2014, pp. 35–36.

[10] P. Tonella, "Evolutionary testing of classes," in *Proceedings of the 2004 ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA '04)*. Boston, Massachusetts, USA: ACM, 11-14 July 2004, pp. 119–128.

[11] J. Wegener and O. Bühler, "Evaluation of different fitness functions for the evolutionary testing of an autonomous parking system," in *Genetic and Evolutionary Computation Conference (GECCO 2004)*, Seattle, Washington, USA, Jun. 2004, pp. 1400–1412, LNCS 3103.

[12] J. Wegener and M. Grochtmann, "Verifying timing constraints of real-time systems by means of evolutionary testing," *Real-Time Systems*, vol. 15, no. 3, pp. 275 – 298, 1998.

[13] J. Oh, M. Harman, and S. Yoo, "Transition coverage testing for Simulink/Stateflow models using messy genetic algorithms," in *Genetic Algorithms and Evolutionary COmputation Conference (GECCO 2011)*, Dublin, Ireland, 2011, pp. 1851–1858.

[14] N. Tracey, J. Clark, K. Mander, and J. McDermid, "Automated test-data generation for exception conditions," *Software Practice and Experience*, vol. 30, no. 1, pp. 61–79, 2000.

[15] M. B. Cohen, P. B. Gibbons, W. B. Mugridge, and C. J. Colbourn, "Constructing test suites for interaction testing," in *Proceedings of the 25th International Conference on Software Engineering (ICSE-03)*. Piscataway, NJ: IEEE Computer Society, May 3–10 2003, pp. 38–48.

[16] Y. Zhan and J. A. Clark, "Search-based mutation testing for simulink models," in *Genetic and Evolutionary Computation Conference (GECCO 05)*, H.-G. Beyer and U.-M. O'Reilly, Eds. Washington DC, USA: Association for Computer Machinery, Jun. 2005, pp. 1061–1068.

[17] S. Yoo, M. Harman, P. Tonella, and A. Susi, "Clustering test cases to achieve effective and scalable prioritisation incorporating expert knowledge," in *ACM ISSTA 09*, Chicago, Illinois, USA, 19th – 23rd July 2009, pp. 201–212.

[18] L. C. Briand, Y. Labiche, and M. Shousha, "Stress testing real-time systems with genetic algorithms," in *Genetic and Evolutionary Computation Conference, GECCO 2005, Proceedings, Washington DC, USA, June 25-29, 2005*. ACM, 2005, pp. 1021–1028.

[19] Y. Zhang, M. Harman, and A. Mansouri, "The SBSE repository: A repository and analysis of authors and research articles on search based software engineering," 2017, http://crestweb.cs.ucl.ac.uk/resources/sbse_repository/.