Overview of TASE 2012 talk on Search Based Software Engineering

Mark Harman
CREST Centre, University College London, Malet Place, London, WC1E 6BT, UK.

***Abstract*—This is an overview of the keynote presentation on SBSE at the Sixth IEEE International Symposium on Theoretical Aspects of Software Engineering (TASE 2012), held on the 4th-6th July 2012 in Beijing, China.**

Search Based Software Engineering (SBSE) is the name given to a field of research and practice in which computational search and optimisation techniques are used to address problems in Software Engineering [1]. This has proved to be a widely applicable and successful approach, with many applications right across the full spectrum of activities in software engineering.

The approach has produced important research results and, more recently, has witnessed more widespread uptake within industry [2], [3], [4], [5], [6], [7]. There are also a number of tools for SBSE applications including tools for testing [8], [9], [10], [11] modularisation [12], and bug fixing [13].

This talk aimed to provide an overview of SBSE, its foundations and motivation, illustrated by some applications and findings from recent studies and concluding with a look ahead to future challenges and opportunities. There are many surveys, overviews and reviews on SBSE which provide a wealth of material on SBSE and its applications and so there is no need for a further paper to provide such an overview at this stage. The author's position on future developments in SBSE towards more dynamic adaptive automated software engineering are described in a forthcoming keynote paper at the sixth ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM 2012) [14].

Readers interested in finding out more about SBSE will find excellent surveys and overviews covering requirements [15], predictive modelling [16], [17], non-functional properties [18], program comprehension [19], design [20] and testing [18], [21], [22], [23]. There is also a recent bibliometric analysis of ten years' of SBSE literature (2001-2010) [24]. An overview of the use of Evolutionary Computation on Software Engineering [25] and a more general position paper on the use of Artificial Intelligence techniques in Software Engineering [26] can also be found elsewhere. Recent surveys on regression testing [27] and mutation testing [28] also contain sections on search based aspects of these two areas of literature.

Those new to SBSE seeking a more general introductory text may like to consult the recent tutorial paper [29]. The tutorial seeks to take the reader from the position of no previous knowledge of SBSE to a point at which he or she is ready to undertake their first experiments. It gives advice on techniques, applications approaches as well as the conduct of experiments and the publication of results on SBSE.

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] W. Afzal, R. Torkar, R. Feldt, and G. Wikstrand, "Search-based prediction of fault-slip-through in large software projects," in *Second International Symposium on Search Based Software Engineering (SSBSE 2010)*, Benevento, Italy, 7-9 Sep. 2010, pp. 79–88.

[3] C. Cadar, P. Godefroid, S. Khurshid, C. S. Păsăreanu, K. Sen, N. Tillmann, and W. Visser, "Symbolic execution for software testing in practice: preliminary assessment," in 33^{rd} *International Conference on Software Engineering (ICSE'11)*. New York, NY, USA: ACM, 2011, pp. 1066–1071.

[4] S. L. Cornford, M. S. Feather, J. R. Dunphy, J. Salcedo, and T. Menzies, "Optimizing Spacecraft Design - Optimization Engine Development: Progress and Plans," in *Proceedings of the IEEE Aerospace Conference*, Big Sky, Montana, March 2003, pp. 3681–3690.

[5] K. Lakhotia, N. Tillmann, M. Harman, and J. de Halleux, "FloPSy — Search-based floating point constraint solving for symbolic execution," in 22^{nd} *IFIP International Conference on Testing Software and Systems (ICTSS 2010)*, Natal, Brazil, November 2010, pp. 142–157, LNCS Volume 6435.

[6] S. Yoo, R. Nilsson, and M. Harman, "Faster fault finding at Google using multi objective regression test optimisation," in 8^{th} *European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE '11)*, Szeged, Hungary, September 5th - 9th 2011, industry Track.

[7] 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.

[8] N. Alshahwan and M. Harman, "Automated web application testing using search based software engineering," in 26^{th} *IEEE/ACM International Conference on Automated Software Engineering (ASE 2011)*, Lawrence, Kansas, USA, 6th - 10th November 2011, pp. 3 – 12.

[9] G. Fraser and A. Arcuri, "Evosuite: automatic test suite generation for object-oriented software," in 8^{th} *European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE '11)*. ACM, September 5th - 9th 2011, pp. 416–419.

[10] Y. Jia and M. Harman, "Milu: A customizable, runtime-optimized higher order mutation testing tool for the full C language," in 3^{rd} *Testing Academia and Industry Conference - Practice and Research Techniques (TAIC PART'08)*, Windsor, UK, August 2008, pp. 94–98.

[11] K. Lakhotia, M. Harman, and H. Gross, "AUSTIN: A tool for search based software testing for the C language and its evaluation on deployed automotive systems," in 2^{nd} *International Symposium on Search Based Software Engineering (SSBSE 2010)*, Benevento, Italy, September 2010, pp. 101 – 110.

[12] B. S. Mitchell and S. Mancoridis, "On the automatic modularization of software systems using the bunch tool," *IEEE Transactions on Software Engineering*, vol. 32, no. 3, pp. 193–208, 2006.

[13] C. Le Goues, T. Nguyen, S. Forrest, and W. Weimer, "GenProg: A generic method for automatic software repair," *IEEE Transactions on Software Engineering*, vol. 38, no. 1, pp. 54–72, 2012.

[14] M. Harman, "Dynamic adaptive search based software engineering," in 6^{th} *IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM 2012)*, Lund, Sweden, 2012.

[15] Y. Zhang, A. Finkelstein, and M. Harman, "Search based requirements optimisation: Existing work and challenges," in *International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'08)*, vol. 5025. Montpellier, France: Springer LNCS, 2008, pp. 88–94.

[16] W. Afzal and R. Torkar, "On the application of genetic programming for software engineering predictive modeling: A systematic review," *Expert Systems Applications*, vol. 38, no. 9, pp. 11 984–11 997, 2011.

[17] M. Harman, "The relationship between search based software engineering and predictive modeling," in 6^{th} *International Conference on Predictive Models in Software Engineering*, Timisoara, Romania, 2010.

[18] W. Afzal, R. Torkar, and R. Feldt, "A systematic review of search-based testing for non-functional system properties," *Information and Software Technology*, vol. 51, no. 6, pp. 957–976, 2009.

[19] M. Harman, "Search based software engineering for program comprehension," in 15^{th} *International Conference on Program Comprehension (ICPC 07)*. Banff, Canada: IEEE Computer Society Press, 2007, pp. 3–13.

[20] O. Räihä, "A survey on search–based software design," *Computer Science Review*, vol. 4, no. 4, pp. 203–249, 2010.

[21] S. Ali, L. C. Briand, H. Hemmati, and R. K. Panesar-Walawege, "A systematic review of the application and empirical investigation of search-based test-case generation," *IEEE Transactions on Software Engineering*, pp. 742–762, 2010.

[22] M. Harman, "Open problems in testability transformation," in *1st International Workshop on Search Based Testing (SBT 2008)*, Lillehammer, Norway, 2008, keynote paper.

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

[24] F. G. Freitas and J. T. Souza, "Ten years of search based software engineering: A bibliometric analysis," in 3^{rd} *International Symposium on Search based Software Engineering (SSBSE 2011)*, 10th - 12th September 2011, pp. 18–32.

[25] M. Harman, "Software engineering meets evolutionary computation," *IEEE Computer*, vol. 44, no. 10, pp. 31–39, Oct. 2011.

[26] ——, "The role of artificial intelligence in software engineering," in 1^{st} *International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE 2012)*, Zurich, Switzerland, 2012.

[27] S. Yoo and M. Harman, "Regression testing minimisation, selection and prioritisation: A survey," *Journal of Software Testing, Verification and Reliability*, vol. 22, no. 2, pp. 67–120, 2012.

[28] Y. Jia and M. Harman, "An analysis and survey of the development of mutation testing," *IEEE Transactions on Software Engineering*, vol. 37, no. 5, pp. 649 – 678, September–October 2011.

[29] M. Harman, P. McMinn, J. Souza, and S. Yoo, "Search based software engineering: Techniques, taxonomy, tutorial," in *Empirical software engineering and verification: LASER 2009-2010*, B. Meyer and M. Nordio, Eds. Springer, 2012, pp. 1–59, LNCS 7007.