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, reviews and overviews 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 [19], [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 [24] and mutation testing [23] 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


