Position Statement on the Role and Future of Search Based Software Engineering

Norman Fenton

SSBSE 2009
1st International Symposium on Search Based Software Engineering
Windsor, UK

13-15 May 2009

During the 1970's and 1980s there were many computer scientists - and software engineers especially - who dreamed of perfect solutions to complex problems based purely on mathematical semantics and logic. With the right formal notation and proof rules they imagined that it would be possible, for example, to create complex software systems automatically from mathematical specifications that were perfect in the sense of requiring no testing [1]. Although for many this dream has never died, and examples of modestly complex, verified systems and tools to support them have been developed, in practice most software engineering challenges can never support perfect solutions [2]. Since even a perfectly verified system can only have been verified against a formal specification, the need for testing still remains for such a system since there can never be a formal automated means of verifying real-world requirements.

The quest for perfect solutions is, of course, also provably infeasible for a wide class of algorithmic problems, namely all of those such as the traveling salesman problem with complexity NP-complete or worse. Since such problems are pervasive and will not just go away, we have been forced to consider solutions that were less than perfect. This led to the broad discipline of approximation in algorithm design with different approaches to optimization and also probabilistic solutions [3]. It seems to me that the current discipline of search-based software engineering [4] is the inevitable extension of this work into areas beyond algorithm design, with a special emphasis on testing. The growing popularity of search-based software engineering therefore seems to be a sign of the maturity of the entire software engineering discipline as it finally moves away from the naivety of those earlier years and confronts the reality of accepting uncertainty and approximation.

While the need for search-based techniques to support testing of complex systems seems to me to be self-evident, my view is that the subject as a whole may be cast in too narrow a light, with too much focus on specific types of metaheuristic search [5]. I can understand why, to date the emphasis has been so. The need for optimising test data is both an obviously important problem and a natural candidate for the class of metaheuristic search techniques commonly used. However, as is made clear in [4] the challenges of search-based software engineering are far greater and could encompass a much broader range of methods. Search-based software engineering essentially deals with any software engineering problem for which there is no perfect solution and for which no efficient deterministic algorithm will find an optimal solution. Such problems are inevitably characterized by uncertainty, and I feel there is a much broader class of methods, which can be classified as 'intelligent', that may be

relevant. At this point I will declare a blatant interest in one such method - Bayesian networks (BNs)– that has been increasingly used to address decision problems involving uncertainty in software engineering [6,7,8,9]. In my background reading for this position statement I was actually surprised to discover that BNs comfortably satisfied the two key ingredients of search-based optimization for a range of software engineering problems, namely:

1. Choice of the representation of the problem
2. Definition of the fitness function

Moreover, BN solutions are also addressing some of the key open problems and challenges in optimization; for example, recent developments on dynamic discretisation algorithms for BNs partially solve the general challenge of stopping criteria cited in [4], while BNs are particularly powerful for addressing multi-objective optimization and sensitivity analysis. The latter are subjects which [4] cites as being part of the road map for future work.

Given that BNs can so easily fit into the criteria (and rationale) for search-based software engineering (while never having formally been considered a candidate), I suspect that there are many other methods beyond the narrow class of metaheuristic search which fit just as well. The big question is: do the current community of search-based software engineering researchers want to move beyond their current boundaries? There are always both benefits and disadvantages of a narrow focus.

References

[1] Hoare, C. A. R. (1986). Maths adds safety to computer programs. New Scientist, pp 53--56, Sept.

[2] Glass, R. L. (2002). "The proof of correctness wars." Commun. ACM **45**(8): 19-21.

[3] Harel, D. (1992). Algorithmics, Addison Wesley

[4] Harman, M., *The Current State and Future of Search Based Software Engineering*, in *2007 Future of Software Engineering*. 2007, IEEE Computer Society. p. 342-357.

[5] Harman, M. and Jones, B. (2001). SEMINAL: Software Engineering using Metaheuristic INnovative ALgorithms. 23rd IEEE/ACM International Conference on Software Engineering (ICSE 2001). Toronto, Canada, May 12th-13th, 2001**:** 762-763.

[6] de Melo, A. C. V. and Sanchez, A. J. (2008). "Software maintenance project delays prediction using Bayesian Networks." Expert Syst. Appl. **34**(2): 908-919, from http://dx.doi.org/10.1016/j.eswa.2006.10.040.

[7] Fenton, N. E., Marsh, W., Neil, M., Cates, P., Forey, S. and Tailor, M. (2004). Making Resource Decisions for Software Projects. 26th International Conference on Software Engineering (ICSE2004) Edinburgh, United Kingdom, IEEE Computer Society**:** 397-406.

[8] Fenton, N. E., et al. (2008). "On the effectiveness of early life cycle defect prediction with Bayesian Nets." Empirical Software Engineering **13**: 499-537.

[9] Stamelosa, I., Angelisa, L., Dimoua, P. and Sakellaris, P. (2005). "On the use of Bayesian belief networks for the prediction of software productivity." Information and Software Tech **45** (1): 51-60.

[10] Neil, M., Tailor, M. and Marquez, D. (2007). "Inference in hybrid Bayesian networks using dynamic discretization." Statistics and Computing **17**(3): 219-233, from http://dx.doi.org/10.1007/s11222-007-9018-y.