Information and Software Technology 78 (2016) 66–82

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Tester interactivity makes a difference in search-based software testing: A controlled experiment

Bogdan Marculescu[a,*], Simon Poulding[a], Robert Feldt[a], Kai Petersen[a], Richard Torkar[b]

[a] Blekinge Institute of Technology, Karlskrona, Sweden
[b] Chalmers and the University of Gothenburg, Gothenburg, Sweden

ARTICLE INFO

Article history:
Received 11 December 2015
Revised 2 May 2016
Accepted 30 May 2016
Available online 31 May 2016

Keywords:
Search-based software testing
Interactive search-based software testing
Controlled experiment

ABSTRACT

Context: Search-based software testing promises to provide users with the ability to generate high quality test cases, and hence increase product quality, with a minimal increase in the time and effort required.

The development of the Interactive Search-Based Software Testing (ISBST) system was motivated by a previous study to investigate the application of search-based software testing (SBST) in an industrial setting. ISBST allows users to interact with the underlying SBST system, guiding the search and assessing the results. An industrial evaluation indicated that the ISBST system could find test cases that are not created by testers employing manual techniques. The validity of the evaluation was threatened, however, by the low number of participants.

Objective: This paper presents a follow-up study, to provide a more rigorous evaluation of the ISBST system.

Method: To assess the ISBST system a two-way crossover controlled experiment was conducted with 58 students taking a Verification and Validation course. The NASA Task Load Index (NASA-TLX) is used to assess the workload experienced by the participants in the experiment.

Results: The experimental results validated the hypothesis that the ISBST system generates test cases that are not found by the same participants employing manual testing techniques. A follow-up laboratory experiment also investigates the importance of interaction in obtaining the results.

In addition to this main result, the subjective workload was assessed for each participant by means of the NASA-TLX tool. The evaluation showed that, while the ISBST system required more effort from the participants, they achieved the same performance.

Conclusions: The paper provides evidence that the ISBST system develops test cases that are not found by manual techniques, and that interaction plays an important role in achieving that result.

1. Introduction

Software testing plays a crucial role in increasing the quality of software systems, as well as the perceived quality of and confidence in such systems. One software testing technique is the application of metaheuristic optimization algorithms to generate test data, known as Search-Based Software Testing (SBST) [1,2].

In a previous study [3], we have proposed a system that would allow successful application of SBST in an industrial context. This system, called the Interactive Search-Based Software Testing (ISBST) tool, facilitated the use of domain knowledge existing in the company to improve the search process. This was achieved by allowing human testers to interact with the system and guide the evolution of the search-based solutions. The interaction was inspired by work in Interactive Evolutionary Computation [4–8], and was designed to allow the testers to make their contribution, without having to deal with the complexity of the underlying SBST system.

Previous work [9] focused on successfully applying ISBST in an industrial context, and determining what were the important factors that enabled successful application. One of the findings of that study was that the ISBST tool developed test cases that were quite different from those obtained by means of manual techniques. However, the evaluation was conducted with a low number of

* Corresponding author.
E-mail address: bogdan.marculescu@bth.se (B. Marculescu).

http://dx.doi.org/10.1016/j.infsof.2016.05.009
0950-5849/

participants and in a context specific to our industrial partner, thus making it difficult to draw conclusions about the ISBST.

This study validates those findings, by conducting a large, controlled experiment, comparing the test cases developed using the ISBST system with those developed using a manual black-box technique. The experiment was conducted with 58 software engineering students, participants in a software Verification and Validation course at the master level. By selecting a more general SUT, in this case a clustering algorithm that is not tied to a particular problem domain or company, we can increase the level of confidence in the generalizability of our method. Master's students are a good choice of participant, as they are not likely to be influenced by the biases and assumptions inherent in any domain, and more will have the time and willingness to participate in an experiment.

The experiment provides evidence that the automated system, represented by the ISBST tool, develops different test cases from the manual method. A follow-up computer-based experiment also provides evidence for the role of interaction in obtaining the results. By isolating the interaction strategy and comparing against the same search-based system without the benefit of interaction, we were able to provide evidence that interaction plays a significant role in the results obtained by the ISBST tool.

The contributions of this paper are as follows:

- Comparing test cases developed by the ISBST system and those developed by manual exploratory testing, to identify differences and similarities between them, and to determine whether or not they investigate the same type of SUT behavior.
- Assessing the effect of the interaction component of the ISBST system on the outcome of the search.
- Widening the application of the ISBST system to a completely new type of System under Test (SUT), part of a different domain.
- Evaluating the ISBST system on a wider set of participants, in a controlled environment.

Section 2 describes existing work on evolutionary approaches and search-based software testing and discusses the context of the current approach, as well as providing a description of the ISBST system itself. In Section 3 we describe the design of the current experiment and the tools used during the empirical process. Sections 4 and 5 present the results from the experiment and discuss their significance, respectively. The threats to the validity of the study are discussed in Section 6, and Section 7 concludes the paper.

2. Context

This experiment is inspired by results from a study conducted with our industrial partner, to investigate the possibility of using interactive search-based software testing (ISBST) to improve the testing process. Our industrial partner develops embedded software for industrial applications. The ISBST tool was previously developed and evaluated in that context, on a small number of company engineers. Therefore, this study will evaluate the ISBST tool outside of that specific context and with a larger number of participants.

We define a "domain specialist" as a person that develops and tests software for their specific domain as part of their activities, but that is not a software engineer. To assist domain specialists, tools are specifically designed to use the terminology, symbols, and concepts specific to the domain, rather than those specific to software development and testing. Thus, they focus on domain experience and expertise rather than knowledge specific to software testing.

In previous work [3] we proposed a tool, called the Interactive Search-Based Software Testing (ISBST) tool, that would use search-based techniques to help in the testing process. It is difficult to develop *a priori* a fitness function that would be useful for a general SUT. As a result, the ISBST tool was designed to use a Dynamically Adapted Fitness Function (DAFF). In this concept, the fitness function is composed of a set of dimensions relevant to system quality to assess each candidate solution. By changing the relative importance of these attributes, the domain specialist can change the fitness function and indirectly guide the search. In our previous study, the relevant dimensions were identified and validated in collaboration with our industrial partner.

Further work [9] resulted in a practical implementation of the ISBST tool. The tool, and the concept of a Dynamically Adapted Fitness Function, were validated in a small case study conducted in an industrial setting. One of the results of that study was that the test cases that were developed by using the ISBST tool were useful and unexpected. The domain specialists using the tool stated that they would not have considered investigating that type of behavior, but that the behavior itself was a good addition to the test suite.

The results of the exploratory study mentioned above indicated that using the ISBST tool would enable domain specialists to guide the search towards a more diverse set of behaviors than they could develop by using manual techniques. The more diverse set of behaviors would then be assessed by the domain specialists, who would refine relevant test cases and add them to the test suites.

Henceforth, we define the "behavior" as the set of measured outputs, or any function of those outputs, corresponding to a given set of inputs of the system under test (SUT). Thus, the "observed behavior space", or just "behavior space", is the total set of possible behaviors for a given SUT. Note that the behavior space deals only with characteristics of the SUT that are measured or evaluated, and is not a complete description of the SUT. The behaviors that are measured and form the behavior space will be called "behavior attributes". The ISBST system may try to optimize, i.e. minimize or maximize, the found values for a given behavior attribute in a direction. In this case, a "search objective" is defined as the combination of behavior attribute and direction.

Additional behavior attributes may be identified and added, if they are considered relevant, and this would result in changes to the behavior space of the SUT. This further complicates attempts to explore the behavior space. For this paper, we define a "test case" to consist of a set of inputs and the corresponding SUT behavior.

The behavior space of a system is, in general, difficult to define and difficult to explore purposefully. Varying only certain characteristics of the behavior is, for most systems, a complex problem. The ISBST tool aims to use system behavior to measure the fitness of a given test case. By doing so, the ISBST tool can explore the behavior space of a system indirectly and develop test cases that explore previously unexercised, and unknown, regions of the behavior space.

2.1. Related work

Search-based software testing (SBST) is the application of metaheuristic optimization methods to the problem of software testing. SBST is part of the larger scope of search-based software engineering, a term coined by Harman and Jones [10]. SBST has been successfully applied on a wide range of software testing problems. McMinn [1] describes the use of SBST for temporal, structural, and functional testing, while Afzal et al. [2] focus their review on the use of SBST on non-functional testing.

Search-based techniques, both in the wider area of software engineering and, more specifically in the field of testing, rely on having an automated means of assessing the quality, or "fitness" of a candidate solution.

However, the definition and understanding of what fitness is, and what candidates are preferable, can change during the search.