ARTICLE IN PRESS

Information Sciences xxx (2010) xxx–xxx

Contents lists available at ScienceDirect

Information Sciences

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

System regression test planning with a fuzzy expert system

Zhiwei Xu [a,*], Kehan Gao [b], Taghi M. Khoshgoftaar [c], Naeem Seliya [a]

[a] University of Michigan-Dearborn, Dearborn, MI, USA
[b] Eastern Connecticut State University, Willimantic, CT, USA
[c] Florida Atlantic University, Boca Raton, FL, USA

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Fuzzy expert systems
System regression test
Fuzzy reasoning

ABSTRACT

An effective system testing activity for consecutive releases of very large software systems depends considerably on the selection of test cases for execution. From a practical point of view, it is not feasible to run all possible test cases because of the real-world constraints of limited time, human and monetary resources. While existing techniques for test case selection provide methods to limit the growth of the number of test cases with software evolution, many of them are based on the assumption that source code analysis is available. Very few works have explored the problem of test case selection given the constraint that source code analysis is not available. We propose fuzzy expert systems as an effective solution to the problem. Fuzzy expert systems have the ability to emulate fuzzy human reasoning and judgment processes. The proposed fuzzy expert system identifies potentially critical test cases for system test by correlating knowledge represented by one or more of the following: customer profile, analysis of past test case results, system failure rate, and change in system architecture. We piloted this fuzzy expert system in a large telecommunications system and the results show that test effectiveness and efficiency is significantly improved.
Published by Elsevier Inc.

1. Introduction

Software maintenance activities can account for as high as two-thirds of the total cost of software production and operation [10,11]. When software is modified for any reason, the system must be thoroughly tested to verify the modifications and to also ensure the changes did not cause any adverse effects on other functionality and requirements of the software. Known as regression testing, system testing after software changes is a necessary and often very expensive software maintenance task.

Among the different strategies for regression testing, one approach is to rerun all existing test cases in the project's test suite. However, this is often prohibitive because of the costs associated with monetary, human, and schedule resources. In addition, working with a random selection of a subset of test cases can be unreliable and ineffective, especially if it leads to no or very few defects being detected. Software practitioners have presented techniques for regression test case selection in an attempt to reduce the cost and time for regression testing by selecting critical test cases from the existing test suite.

Several regression test case selection techniques [3,4,19,20] are based upon analyzing information contained in the source code of the program and any of its modified versions. However, software engineering practice often deals with scenarios where the source code is either not available or cannot be made accessible to the system testing team. The growing trend of distributed software development and outsourcing often formulates a software development team in which the developers and system testers are at separate remote locations. Consequently, security and information privacy issues may prevent the system testing team from accessing the source code.

* Corresponding author.
E-mail address: zwxu@umich.edu (Z. Xu).

0020-0255/$ - see front matter Published by Elsevier Inc.
doi:10.1016/j.ins.2010.09.012

Development organizations that implements well-known software testing practices as recommended by independent verification and validation processes often face the problem of source code unavailability at the system (black-box) testing level. Considering this problem, Xu and Samaan [27] report on the importance of correlating the user operational profile with the change in software architecture and the analysis of previous test results. They investigate this correlation using a simple logic approach in an attempt to optimize test cases selection for regression testing. However, they also emphasize the limitation of always using simple logic, particularly in problematic situations such as, when the number of nominated test cases for regression testing is much greater than what is acceptable as cost effective. To address this problem, we propose using a fuzzy expert system to address this issue.

When the test suite for a given project is relatively small, an experienced tester can effectively identify the critical test cases by analyzing various attributes/information from the software project and testing knowledge base. Such knowledge for analysis can include the software requirements document, test case behaviors in the previous system releases, customer profiles, and the change in system architecture. However, when the test suite is relatively large, it is either not feasible or practically impossible for the system testing team to review the entire pool of test cases before making an informed decision about constituting a set of critical test cases.

The primary objective of this study is to provide the system testing team with a fuzzy expert system that can be used during the decision-making process of regression test case selection when the test suite is large. The proposed expert system emulates the often imprecise human reasoning and judgment process by correlating knowledge represented by one or more of the following project- and testing-related information: customer profile, analysis of prior test case results, system failure rate, and changes in system architecture. The outcome of the correlated analysis by the expert system is an identified set of potentially critical test cases for regression testing.

We piloted the proposed fuzzy expert system in a large telecommunications software system. The process involved interviewing the individual system test managers in order to extract high-level rules for our expert system. These high-level rules were then transformed into more detailed rules that can be recognized by the fuzzy expert system. The formulation and application of a fuzzy expert system for test case selection during system level regression testing are unique to this study. The fuzzy expert system is evaluated based on data obtained from a real-world test case suite of the large telecommunications system. Our empirical results indicate the proposed fuzzy expert system is effective in emulating human judgment in the identification of the critical test cases.

The remainder of the paper is organized as follows. In Section 2, we discuss some related literature. In Section 3, the test case selection process based on the fuzzy expert system are presented. In Section 4, the real-world case studies used in this study are described. Finally, in Section 5, we summarize the work presented in this paper along with suggestions for relevant future work.

2. Related work

To reduce the cost of regression test, many researchers have conducted various studies on test case selection or prioritization [2–4,19,20,23]. Rothermel et. al. use several methods to prioritize the test cases, such that the most important test cases will be run relatively earlier in the regression testing process [3–5,8,19,20]. Researchers have shown that test case prioritization techniques can improve a test suite's rate of fault detection.

Last et al. [12] demonstrated the potential use of data mining algorithms for automated induction of functional requirements from execution data. The induced data mining models of tested software can be utilized for recovering missing and incomplete specifications, designing a minimal set of regression tests, and evaluating the correctness of software outputs when testing new, potentially flawed releases of the system. The researchers applied a data mining algorithm called Info-Fuzzy Network (IFN) to the execution data of a general-purpose code for solving partial differential equations. The results show that the model constructed by the IFN algorithm has the capability of discriminating between correct and faulty versions of the program.

Srivastava and Thiagarajian presented the "Echelon" test prioritization system at Microsoft [23]. The system inputs two versions of the program in binary form and test coverage information of the older version, and outputs an ordered list of test sequence. Echelon is able to operate on large binaries built from millions of lines of source code and produce results within a few minutes. Harrold et al. conducted a study that uses component metadata for regression selection of COTS components [15]. They used metacontents for code-based regression test selection techniques. Their case study showed that on average 26% of the overall testing effort was saved over seven releases.

Zheng developed a process called the Integrated-Black-box Approach for Component Change Identification (I-BACCI) process [29]. The input artifacts are the binary code of the components (old and new versions), the source code, and test suite of the application (including the glue code). Once the process is completed, the reduced set of regression test cases can be run to determine if any of the changes in the COTS components affected the operation of the application. Once the changed components are identified, the I-BACCI process utilizes the Firewall model of regression test selection [29].

Li et al. investigated five search techniques for regression test case prioritization: two metaheuristic search techniques (Hill Climbing and Genetic Algorithms), together with three greedy algorithms (Greedy, Additional Greedy, and 2-Optimal Greedy) [13]. The paper presents results from an empirical study that compared the performance of the five search

Please cite this article in press as: Z. Xu et al., System regression test planning with a fuzzy expert system, Inform. Sci. (2010), doi:10.1016/j.ins.2010.09.012