Identifying Categories and Choices for Software Testing Based on Informal Specifications

Sau Fun TANG

Submitted in total fulfilment of the requirements

of the degree of Doctor of Philosophy

March 2009

Faculty of Information and Communication Technologies

Swinburne University of Technology

Abstract

It would be ideal to test a program with all its possible inputs (that is, the *input domain*). In this way, any fault that may exist in the program would be guaranteed to be detected. In practice, however, this "exhaustive" approach is difficult, or almost impossible, to apply because of limited testing resources. Therefore, a more feasible approach is to construct a *test suite*, which is a subset of the input domain, for testing. The fundamental problem of this approach is how to construct a test suite so that the comprehensiveness of the test cases, as compared to the input domain, is jeopardized as less as possible.

To solve this problem, two closely related test suite generation techniques have been developed and have attracted much attention. They are the **CHO**i**C**e re**LAT**ion fram**E**work (abbreviated as CHOC'LATE) and the **C**lassification-**T**ree **M**ethodology (abbreviated as CTM). A main advantage of both techniques is that they can be applied to informal specifications, which are commonly used in the commercial software industry.

In CHOC'LATE and CTM, an early step is to identify a set of categories and choices from the specification, through which a test suite is generated. This set of categories and choices is very important because it affects the comprehensiveness of the test suite, which in turn affects the chance of detecting software faults. We observe that neither the original developers of CHOC'LATE/CTM nor follow-up researchers have proposed a systematic method for identifying categories and choices from informal specifications. As a result,

the identification process is often performed in an ad hoc, impromptu manner. Without doubt, an impromptu approach cannot assure the quality of the identified categories and choices. This thesis addresses such a problem, with a view to developing an effective methodology for identifying categories and choices from informal specifications.

To provide the background motivation, we conduct two rounds of empirical studies using several commercial specifications primarily written in an informal manner. The first round of studies investigates the common mistakes made by *inexperienced* software testers in an impromptu identification approach. This round of studies serves three purposes: (a) to make the knowledge of common mistakes known to other testers so that they can avoid repeating the same mistakes, (b) to facilitate researchers and practitioners develop systematic identification methodologies, and (c) to provide a means of measuring the effectiveness of newly developed identification methodologies. We also present a checklist to help testers detect the common mistakes. In the second round of studies, we shift our focus to *experienced* testers with many years of commercial experience in software development and testing. The objectives of the studies are: (a) to investigate the differences in the types and amounts of mistakes made between inexperienced and experienced testers in an impromptu identification approach, and (b) to determine, after discussing the mistakes with the testers and providing them with a checklist as a simple guideline for detecting problematic categories and choices, the extent of mistake reduction in the next identification exercise. We also discuss which specification components are useful for category and choice identification.

Thereafter, we present the most significant work of the thesis: our **D**ivid**E**-and-conquer methodology for identifying categorie**S**, choice**S**, and choic**E** **R**elations for **T**est case generation (abbreviated as DESSERT). The methodology is particularly useful for informal specifications which are complex and contain many different types of specification components. The theoretical framework and the associated algorithms of DESSERT will

be discussed in detail. We shall also describe two case studies using commercial specifications to demonstrate the effectiveness of DESSERT.

Acknowledgement

Firstly, I wish to express my gratitude to my coordinating supervisor, Prof. T. Y. Chen, and associate supervisors, Prof. T. H. Tse and Dr. F.-C. Kuo, who helped me throughout this work by providing constant guidance, support, and stimulating feedback.

Acknowledgement is due to the Department of Computer Science of The University of Hong Kong, where part of my research was performed. I am also indebted to Asso. Prof. Y. T. Yu in the Department of Computer Science at City University of Hong Kong and Asso. Prof. P. L. Poon in the School of Accounting and Finance at The Hong Kong Polytechnic University for their invaluable help and insightful advice. I also thank the Faculty of Information and Communication Technologies at Swinburne University of Technology and their staff for the assistance and support offered to me.

I gratefully acknowledge the financial support from a Swinburne University Postgraduate Research Award during part of my Ph.D. candidature. Part of the work in this thesis was supported by a research grant (project no. 123206) from the Hong Kong Research Grants Council.

Declaration

I hereby declare that this thesis comprises the original work undertaken during my Ph.D. candidature. The thesis contains no material which has been accepted for the award to me of any other degree or diploma. In addition, to the best of my knowledge, the thesis contains no material previously published or written by another person except where due reference is made in the text of the thesis. Parts of this thesis are the results of joint research or collaboration with others [10, 11, 15, 16, 18, 49, 62].

I also certify that the thesis is less than 100 000 words in length, excluding tables, footnotes, and bibliographies.

Signature: _____

Sau Fun Tang

Contents

List of Tables

List of Figures