Available online at www.sciencedirect.com

SCIENCE DIRECT

Information and Software Technology 48 (2006) 645–659

INFORMATION AND SOFTWARE TECHNOLOGY

www.elsevier.com/locate/infsof

A method for assigning a value to a communication protocol test case

Richard Lai[a,*], Yong Soo Kim[b]

[a]Department of Computer Science and Computer Engineering, La Trobe University, Bundoora, Vic. 3086, Australia
[b]College of Software, Kyungwon University, Songnam, Kyunggi-Do 461-701, South Korea

Received 30 August 2004; revised 30 June 2005; accepted 6 July 2005
Available online 18 August 2005

Abstract

One of the main problems in industrial testing is the enormous number of test cases derived from any complex communication protocol. Due to budget constraints and tight schedule, the number of test cases has to be within a certain limit. However, by having a limit on the number of test cases, it raises some issues. For instances, what criteria should be used for selecting the test cases? How can we ensure that important test cases have not been excluded? We are proposing that assigning a value to each of the test cases of a test suite can provide a solution. By doing so, the relative importance of each of the test cases can be ranked and an optimal test suite can then be designed. The value of a test case is to be measured in economic terms, which could be based on the probability that a particular case will occur, and the probability that an error is likely to be uncovered. This paper presents a method for assigning a value to a test case of a communication protocol; it is based on sensitivity analysis, which involves execution, infection and propagation probabilities. To illustrate the method, the results of applying it to the INRES protocol are presented.
© 2005 Elsevier B.V. All rights reserved.

Keywords: Formal Description Technique; Protocol testing; Estelle; Sensitivity analysis; Test case value

1. Introduction

The advance in technology in telecommunications has resulted in the rapid development of many complex communication protocols. Communication protocol testing is playing a more and more crucial role in the development of computer network as it provides a means of enhancing the reliability of communication software. Due to their complexities, it is essential to develop a strategy for effective testing. The advance in protocol specification using Formal Description Techniques (FDTs) in recent years has opened up a new horizon for protocol testing [1]. Internationally standardized FDTs include SDL [2], Estelle [3] and LOTOS [4]. A test sequence for a protocol is a sequence of input–output pairs derived from the protocol specification. It can be generated from a formal specification for conformance testing. Successes have been achieved by academia on automatic test case generation from formal specifications [5,6]. In general, more work has been done in automating test case generation from Estelle specifications [7] then has been done with SDL or LOTOS.

Conformance testing is concerned with the conformance of an implementation under test (IUT) to the standards [8]. International Organisation for Standardisation (ISO) has been working towards defining an abstract testing methodology and a framework for specifying conformance test suites since 1984. The effort resulted in the standard ISO 9646 [9], which defines the details for Conformance Testing Methodology and Framework (CTMF). A test notation called Tree and Tabular Combined Notation (TTCN) [9] has also been developed. TTCN is used so that conformance test suites can be shared among testers. CTMF is a very general framework, which could be applicable to the widest possible range of specifications and products.

Despite the work by ISO and academia, there is still a big gap between testing practice and research results published in journals and reported at conferences [10]. This gap between academic and industrial testing practices and the fact that academia has not been addressing the real-life testing issues and problems account for the fact that academic testing methods are seldom used in industry [10]. Thus, there is a growing need for academic research on communication protocol testing to become more

* Corresponding author. Tel.: +61 3 94792374; fax: +61 3 947993060.
E-mail address: lai@latrobe.edu.au (R. Lai).

industrially relevant in order to help narrow the gap. One of the main problems in industrial testing is the enormous number of test cases derived from any complex communication protocol. It would take many man-years to test all the possible test cases.

Due to budget constraints and tight schedule, the number of test cases has to be within a certain limit. However, by having a limit on the number of test cases, it raises some issues. For instances, what criteria should be used for selecting the test cases? How can we ensure that important test cases have not been excluded? What technique should be used for designing an optimal test suite? We are proposing that assigning a value to each of the test cases of a test suite can provide a solution. By doing so, the relative importance of each of the test cases can be ranked and an optimal test suite can then be designed. The value of a test case is to be measured in economic terms, which could be based on the probability that a particular case will occur, and the probability that an error is likely to be uncovered. For this purpose, the testability and sensitivity of a program need to be analysed and understood.

A program's testability is a prediction of its ability to hide faults when the program is black-box-tested with inputs selected randomly from a particular input distribution [12]. A program has high testability when it readily reveals faults through random black-box testing; a program with low testability is unlikely to reveal faults through random black-box testing. A program with low testability is dangerous because considerable testing may make it appear that the program has no faults when in reality it has many.

'Sensitivity' means a prediction of the probability that a fault will cause a failure in the software at a particular location under a specified input distribution [12]. For instance, if a location has a sensitivity of 0.99 under a particular distribution, almost any input in the distribution that executes the location will cause a program failure. On the other hand, if a location has a sensitivity of 0.01 relatively few inputs from the distribution that executes would cause the program to fail, no matter whether or not faults exist at that location.

This paper presents a method for assigning a value to a test case of a communication protocol; the method uses a software testability technique, called Sensitivity Analysis [13], which is based on the Propagation, Infection, and Execution analysis (PIE) model [11]. The Contest tool [14] is employed to generate test cases from an Estelle [3] specification of a communication protocol; based on the three analyses, each transition of an implementation under test is analyzed; execution, infection and propagation probability can be determined. The value of a test case can then be derived from the number of transitions executed combined with this set of probabilities. To illustrate the method, the results of applying it to the INRES protocol are presented. It is our wish that this paper prompts other researchers to develop different methods for assigning a value to a test case.

2. Related work

A detailed study of formal methods with regard to protocol testing can be found in [1,25], which pointed out that the Unique Input/Output (UIO) test sequence method [26–28] can achieve a high fault coverage. However, some faults are still undetected in some fully specified machines [29]. Therefore, in [29] a comprehensive analysis of fault coverage (including modeling) for completely specified machines has been discussed in order to alleviate the problems described in [1,25]. To estimate fault coverage in [29], a functional fault model is defined, with respect to three types of faults [29]. There have been analytical attempts both to classify faults and to characterize the kinds of faults that a certain conformance test generation procedure can detect [29]. Even though interesting, some of these results are based on imprecise definitions [30], and fault masking should be used to argue the relative merits of various testing methods [29].

Several studies [31–33] have tried to find out how to measure the goodness of a set of test cases and how to generate or select test suites with some good coverage measure. Development and implementation of a test case selection algorithm based on coverage metrics and testing distances between control execution sequences are presented in [32]. Improvement of the approach is presented in [33], in which the definition of the metric in [31] has been improved in order to tackle test selection and test coverage for protocols with parallelism and recursion. A study that is related to fault models is discussed in [34]. The purpose is to show the importance of fault models in testing, and to describe various fault models that correspond to different description techniques, which are used for hardware, software and/or communication protocols. Predicting the number of faults is not always necessary; it may be enough to identify the most trouble some modules [35]. In [35], discriminant analysis has been applied to identify fault-prone modules, and a very large telecommunication system (approximately 1.3 million lines of code) has been used for a case study. This study focuses on non-parametric discriminant analysis rather than the parametric approach in modeling methodology. Information about the reuse of each module from a prior release is significant to software quality models for improving the accuracy of predictions found in [35].

Gotha [36] is a tool for generating an abstract test suite for a finite state machine (FSM) driven by a coverage model. The finite state machine is described in a high level language for modeling concrete systems. Such systems may be hardware architectures or components, software systems, communication protocols, or other complex systems and processes. A test case is a sequence of stimuli for the model. The Gotha prototype was originally developed for hardware architecture models but we have exploited it for modeling and testing software systems. Due to an increasing interest in SDL, MSC and TTCN based tools for validation and test

Download English Version:

https://daneshyari.com/en/article/551564

Download Persian Version:

https://daneshyari.com/article/551564

Daneshyari.com