International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:9, No:10, 2015

Model-Based Software Regression Test Suite Reduction

Shiwei Deng, Yang Bao

***Abstract*—**In this paper, we present a model-based regression test suite reducing approach that uses EFSM model dependence analysis and probability-driven greedy algorithm to reduce software regression test suites. The approach automatically identifies the difference between the original model and the modified model as a set of elementary model modifications. The EFSM dependence analysis is performed for each elementary modification to reduce the regression test suite, and then the probability-driven greedy algorithm is adopted to select the minimum set of test cases from the reduced regression test suite that cover all interaction patterns. Our initial experience shows that the approach may significantly reduce the size of regression test suites.

***Keywords*—**Dependence analysis, EFSM model, greedy algorithm, regression test.

I. Introduction

DURING software maintenance of large software systems, their specification and implementation are changed sometimes to fix defects, to enhance or change functionality, to add new functionality, or to delete the existing functionality. Regression testing is the process to validate that the changes introduced in a system are correct and do not adversely affect the unchanged portion of the system. During regression testing, previously developed test cases are deployed for revalidating a modified system, but also new test cases are frequently generated. Regression testing tends to consume a large amount of time and computing resources, especially for large software systems. To minimize time and effort spent on testing and re-testing a software product, there has been a significant amount of research on the design of effective regression testing techniques to reduce the cost of regression testing [1]-[3].

In recent years, model-based test generation techniques have been developed. These techniques are appropriate for state-based systems that can be modeled using formal description languages like Extended Finite State Machine (EFSM). These techniques are used to automatically generate system-level test suites from system models even for large software systems. The system models are frequently modified to reflect changes in specifications. When the system model is changed, selective regression test generation techniques are used to generate regression test suite (RTS) to test the modified parts of the model. The size of these regression test suites may be very large even for relatively small systems. Therefore, regression test suite reduction is important, especially, in situations executing regression tests is very time and resource consuming for large software systems [4].

In this paper, we present a novel approach of model-based regression test reduction that uses EFSM model dependence analysis and probability-driven greedy algorithm to reduce a given regression test suite. The approach automatically identifies the difference between the original and modified system models by identifying a set of elementary modifications (EMs): elementary addition of a transition and elementary deletion of a transition. For each elementary modification, regression test reduction strategies that use EFSM dependence analysis are used to reduce the regression test suite by eliminating repetitive test cases. The probability-driven greedy algorithm is adopted to select the minimum set of test cases from the reduced regression test suite that cover all interaction patterns (IPs). Our initial experience shows that this approach may significantly reduce the size of regression test suites.

II. Step1: Regression Test Suite Reduction by EFSM Dependence Analysis

The presented regression test suite reduction approach accepts as inputs: the original model, the modified model, and a given regression test suite. The regression test suite may be generated using model-based regression test generation technique. The presented regression test suite reduction approach is independent of the way the regression test suite was originally created. The goal of the approach is to reduce the given regression test suite using EFSM dependence analysis and probability-driven greedy algorithm.

We use an EFSM to model requirements of the system under test. Each requirement of the system under test can be mapped onto the EFSM model. In our discussion, usually, a requirement is specified as a sequence of transitions, and a function is mapped to (a part of) a transition of the EFSM model. Any changes to requirements or functions will be mapped to changes to transitions in the EFSM model. Hence, once an EFSM model is created to represent requirements of a given system under test, testing changed requirements or functions is equivalent to testing changed transitions in the EFSM model. An EFSM is represented as a directed graph where states are represented as nodes and transitions as directed edges between states. In the EFSM model, data dependence (DD) and control dependence (CD) may exist between transitions. Dependence analysis focuses on interactions between transitions in the model. Data dependence captures the notion that one transition defines a value for a variable and the same or some other transition may potentially use this value. Control dependence captures the notion that one transition may influence the traversal of another transition. DDs and CDs in an EFSM model are graphically represented in a Static Dependence Graph (SDG). In SDG, nodes represent EFSM transitions and directed

Shiwei Deng is with the Beijing Institute of System Engineering, P.O.BOX 9702-19, Beijing 100101, China (phone: 8610-64871697; e-mail: s.w.deng@163.com).

Yang Bao is with the Beijing Institute of System Engineering, P.O.BOX 9702-19, Beijing 100101, China (e-mail: baoyang18@163.com).

edges represent EFSM data and control dependencies.

When a model is modified, three types of model-based regression testing need to be performed: 1) Testing the effects of the model on the modification, 2) Testing the effects of the modification on the model, and 3) Testing the side-effects of the modification on the unmodified parts of the model. The goal of these three types of regression testing is to test the different interactions or interaction patterns between functional elements of the model with respect to a transition that represents an elementary modification. Since there are three types of regression testing, three types of IPs related to each EM are introduced:

1) An affecting interaction pattern (Affecting IP),
2) An affected interaction pattern (Affected IP), and
3) A side-effect interaction pattern (Side-effect IP)

The Affecting IP captures interactions between model elements that affect the modification. The Affected IP captures interactions between model elements that are affected by the modification. Finally, the Side-effect IP captures interactions between model elements that occur because of indirect effects introduced by the modification. The original SDG changes accordingly when an original EFSM model is modified. Given a test case, the three IPs can be computed: the Affecting IP, the Affected IP, and the Side-effect IP.

For each given test case, during the traversal of the test case within the modified SDG, up to 3*N IPs are computed, where N is the number of EMs. Two test cases are considered to be equivalent with respect to a transition corresponding to an EM if, during traversal of the EFSM model, these tests exhibit the same IP of certain type with respect to the transition. Only those test cases that at least one of its 3*N IPs does not exist for any other test cases in the reduced test suite are included in the reduced RTS.

In Xie's work, her algorithm takes one EM and calculates IPs for the first test case in a given RTS [5]. If there is an IP calculated, this test case will be selected. All calculated IPs will be categorized by their types (Affecting IP, Affected IP, and Side-effect IP). Then the algorithm takes the next test case and calculates IPs with respect to the EM. This test case will be selected only if a unique IP (has not been calculated before) is calculated. Newly calculated IPs will be categorized by their types, too. The calculation and selection procedure will be repeated until all EMs and all test cases in the given RTS have been processed.

III. Step2: Regression Test Suite Reduction Based on Probability-Driven Greedy Algorithm

In classical greedy algorithm, greedy choice is in effect random. Let $A = \{A_1,...,A_n\}$. The greedy choice can be viewed as implicitly based on a probability assignment on the subsets, such that

$$P(A_i \mid U) > P(A_k \mid U) \Leftrightarrow |A_i \cap U| > |A_k \cap U|, i,k \in \{1,...,n\} \quad (1)$$

In general, this assignment can be improved. If there is only one subset A_i in A that covers an element U_j in U， that subset will be certainly included in any cover of U, not matter what size the subset has. By extension, probability of a subset A_i covering an element U_j to be included in a cover decreases in the number of alternatives, i.e. the number of subset A_k such that A_k covers U_j.

Rampone's probability-driven greedy algorithm formalizes the underlying probability distribution as follows [6]: Let $A = \{A_1,...,A_n\}$ and $U = \{U_1,...,U_m\}$.For each element U_j and each subset A_i, the probability of A_i given U_j is defined as:

$$P(A_i \mid U_j) = IA_i(U_j) / \sum_{k=1}^{n} IA_k(U_j) \quad (2)$$

where $i,k \in \{1,...,n\}$ and $j \in \{1,...,m\}$.

IA_i (U_j) is the characteristic function of the A_i, that is 1 if $U_j \in A_i$ and is 0 otherwise. By taking the mean on U, the probability of A_i with respect to the ground set U is:

$$P(A_i \mid U) = (1/m) \sum_{j=1}^{m} P(A_i \mid U_j) \quad (3)$$

where $i \in \{1,...,n\}$ and $j \in \{1,...,m\}$.

Equation (3) is applied to define a greedy approximation algorithm for set covering problem (SCP) as follows: choose the subset A_i whose probability with respect to U is the maximum, delete the elements of A_i from U, and repeat this process until the ground set U is empty.

We need to select the minimum set of test cases that cover all IPs. This is an SCP problem. We can adopt the probability-driven greedy algorithm as a solution. In Step1, relationships between test case and its construct IP can be tracked in a table as Table I.

TABLE I
Test Cases vs. Covered IPs

	Test Case #1	Test Case #2	.....	Test Case #n
IP_1	0	1	.....	0
IP_2	1	1	.....	0
.....	.....	.....	.....	.....
IP_m	1	0	.....	1

We adapt Rampone's probability-driven greedy algorithm to select the minimum set of test cases from the given RTS that cover all IPs generated in Step1. In our case, the universe U is the set of IPs generated in Step1. Each test case in the given RTS covers several IPs, which is a subset A_i of U. Therefore, the RTS is the family A of subset of U.

Refer to (2), in our test suite reducing approach, U_j is an IP, which is represented by a row of Table I, m is the number of calculated IPs, which equals to the number of rows of Table I. A_i is a test case in the given RTS, which is represented by a column of Table I, n is the number of test cases in the given RTS, which equals to the number of columns of Table I.

IV. Case Study

We use the simplified ATM EFSM model shown in Fig. 1 as an example. There were 5 states and 8 transitions in the original

EFSM model. Suppose the set of EMs to be applied on the EFSM model contained two EMs: m1: adding t_9; m2: deleting t6. Notice that transition $t_{6\ dummy}$ is added to replace the deleted transition to indicate that the Deposit input is consumed in state S_2 and contains an empty sequence of actions. The modified EFSM model is shown in Fig. 2. The given RTS contained 93 test cases.

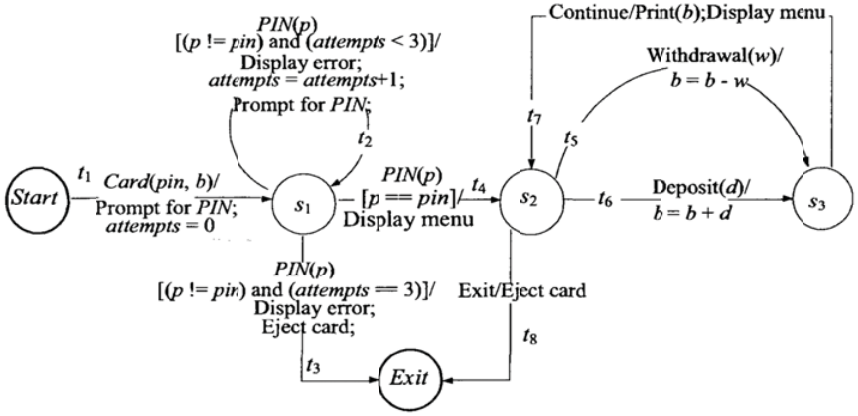

Fig. 1 EFSM model of a simplified ATM system

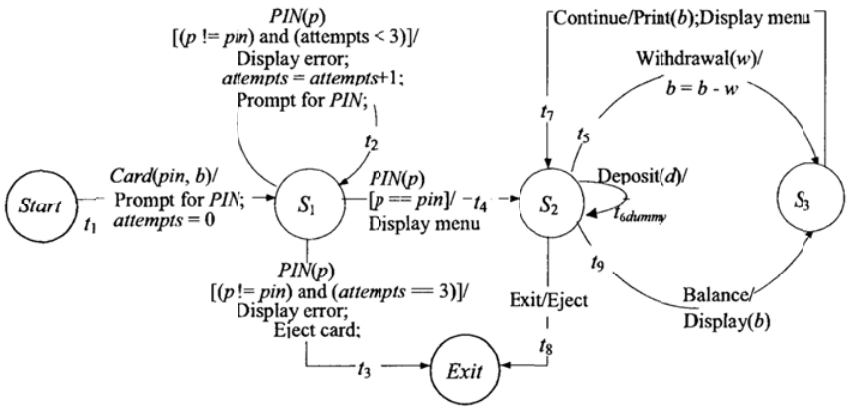

Fig. 2 Modified EFSM model of the simplified ATM system

At Step1, 21 IPs with respect to addition of t_9 and deletion of t_6 are calculated; the 15 test cases contained in reduced RTS are listed in Table II. Relationships between test cases in the reduced RTS and IPs covered by the test cases are obtained. At Step2, the probability-driven greedy algorithm is adopted to solve the test cases selection problem as an SCP problem. After Step2, the 5 test cases contained in reduced RTS are listed in Table III.

The reduced RTS contains only 5.38% of test cases in the given RTS, while Xie's reduced RTS contains 16.13% of the test cases. Chen's reduced RTS contains 8.60% of the test cases, and more time and computing resources are needed [7]. These results show that, for the same EFSM model, same set of EMs and same RTS, our regression test suite reducing method reduced the size of the given RTS more effectively.

V. Conclusion

In this paper, we have presented an approach of model-based regression test reduction. At Step1, the approach identifies the difference between the original model and the modified model as a set of elementary modifications. For each elementary modification, for each test in the regression test suite, interaction patters are identified based on the EFSM dependence analysis. These patterns are used to reduce the regression test suite. At Step2, we adapt probability-driven greedy algorithm to select the minimum set of test cases from the reduced RTS that cover all IPs. Our initial experience shows that the approach may significantly reduce the size of regression test suites.

TABLE II
THE REDUCED RTS AFTER STEP1

Test Case# in Given RTS	Transition Sequences in the Test Case
2	$t_1\ t_4\ t_5\ t_7\ t_9\ t_7\ t_8$
3	$t_1\ t_4\ t_5\ t_7\ t_{6dummy}\ t_9\ t_7\ t_8$
4	$t_1\ t_4\ t_{6dummy}\ t_5\ t_7\ t_8$
5	$t_1\ t_4\ t_{6dummy}\ t_5\ t_7\ t_9\ t_7\ t_8$
6	$t_1\ t_4\ t_{6dummy}\ t_5\ t_7\ t_{6dummy}\ t_9\ t_7\ t_8$
7	$t_1\ t_4\ t_{6dummy}\ t_9\ t_7\ t_5\ t_7\ t_8$
8	$t_1\ t_4\ t_{6dummy}\ t_9\ t_7\ t_5\ t_7\ t_9\ t_7\ t_8$
17	$t_1\ t_4\ t_5\ t_7\ t_5\ t_7\ t_9\ t_7\ t_8$
18	$t_1\ t_4\ t_5\ t_7\ t_5\ t_7\ t_{6dummy}\ t_9\ t_7\ t_8$
19	$t_1\ t_4\ t_5\ t_7\ t_{6dummy}\ t_5\ t_7\ t_8$
20	$t_1\ t_4\ t_5\ t_7\ t_{6dummy}\ t_5\ t_7\ t_9\ t_7\ t_8$
21	$t_1\ t_4\ t_5\ t_7\ t_{6dummy}\ t_5\ t_7\ t_{6dummy}\ t_9\ t_7\ t_8$
33	$t_1\ t_4\ t_{6dummy}\ t_5\ t_7\ t_5\ t_7\ t_{6dummy}\ t_9\ t_1\ t_8$
46	$t_1\ t_4\ t_{6dummy}\ t_9\ t_7\ t_5\ t_7\ t_5\ t_7\ t_8$
47	$t_1\ t_4\ t_{6dummy}\ t_9\ t_7\ t_5\ t_7\ t_5\ t_7\ t_9\ t_7\ t_8$

TABLE III
THE REDUCED RTS AFTER STEP2

Test Case# in Given RTS	Transition Sequences in the Test Case
5	$t_1\ t_4\ t_{6dummy}\ t_5\ t_7\ t_9\ t_7\ t_8$
18	$t_1\ t_4\ t_5\ t_7\ t_5\ t_7\ {}_{t6dummy}\ t_9\ t_7\ t_8$
21	$t_1\ t_4\ t_5\ t_7\ t_{6dummy}\ t_5\ t_7\ t_{6dummy}\ t_9\ t_7\ t_8$
33	$t_1\ t_4\ t_{6dummy}\ t_5\ t_7\ t_5\ {}_{t7}\ t_{6dummy}\ t_9\ t_1\ t_8$
46	$t_1\ t_4\ t_{6dummy}\ t_9\ t_7\ t_5\ t_7\ t_5\ t_7\ t_8$

REFERENCES

[1] E. Engström, P. Runeson, M. Skoglund, "A systematic review on regression test selection techniques," *Information and Software Technology*, vol.52, no.1, 2010, pp.14-30.

[2] M. Salehie, S. Li, L. Tahvildari, R. Dara, S. Li, M. Moore, "Prioritizing Requirements-Based Regression Test Cases: A Goal-Driven Practice," in *Proc. 15th European Conference on Software Maintenance and Reengineering,* Lisbon, Portugal, 2011, pp.329-332.

[3] R. P. Gorthi, A. Pasala, K. K. Chanduka and B. Leong, "Specification-based approach to select regression test suite to validate changed software," in *Proc.15th Asia-Pacific Software Engineering Conference,* Beijing, China, 2008, pp.153-160.

[4] B. Korel, L. H. Tahat, B. Vaysburg, "Model based regression test reduction using dependence analysis," in *Proc. International Conference on Software Maintenance (ICSM2002)*, Montreal, Canada, 2002, pp. 214-223.

[5] B. Xie, *Requirement-based Regression Test Suite Reduction Use Dependence Analysis*, Master's thesis, University of Ottawa, Canada, 2005.

[6] S. Rampone, "Probability-driven Greedy Algorithms for Set Cover," in *Proc. VIII SIGEF Congress "New Logics for the New Economy"*, Naples, Italy, 2001, pp. 215-220.

[7] Y. Chen, *Specification-based Regression Test Suite Generation and Reduction,* Ph.D. thesis, University of Ottawa, Canada, 2009.