Test Case Prioritization Based on Clustering Analysis

Yi-fanXUE[1,a], Yu-guangMAO[1,2]

1.College of Computer Science and Technology,NanjingUniversity of Aeronautics and Astronautics,Nanjing 210016;

2.State Key Laboratory for Nover Software Technology,NanjingUniversity,Nanjing 210093,China

[a]email:xueyifan_nj@163.com

Key words: priority, test case, code, test

Abstract. With regard to the existing test case priority algorithm rarely focusing on the static covering condition of test case and other design information, as well as the testing person need to online collect relevant information about each statement in the procedure, and a higher dependence on code information, and so on, this paper proposes a online adjustment strategy for the priority ranking of test case basing on multiple-objective optimization. Firstly, it is possible to define the requirement coverage matrix of test case basing on the definition of design information of test case. Then, on the basis of the content above mentioned, it introduces three impact factors of test case priority, which are respectively coverage matrix of attention requirement, the importance degree of test case based on requirement and failure rate. In addition, it online estimates the attention requirementcoverage rate and failure rate of test case according to the coverage requirement condition and the results from the execution of the test cases in the testing procedure. Finally, online adjustment of test case priority is realized. The advantage of this method lies in that it keeps the advantage of online adjustment for priority without depending on code information, and avoids the extra costs.

Introduction

With the rapid development of software industry, the frequency of the software product upgrading is higher and higher, which makes the cost of regression testing is higher and higher. In the test of large-scale software, it usually costs several days even months to complete the centralized test by applying complete running of test cases. Under this situation, tester hopes to sort the priority of test case according to certain standard in order to make the case in higher priority could be executed as early as possible, thus to increase the test efficiency. Test case prioritization technique is just proposed to solve this problem. This technique points out that different test cases contribute in different degrees to the completion of test target. In order to rapidly achieve the test target, it is necessary to compare and sort different test cases on the basis of historical test information, consequently to execute relatively important test cases in priority. Interlocking test case generation mainly includes two methods. Literature[3]: the test case is generated from the specification instruction expressed by relational algebra query. Literature [4]: the generation method is based on the test data declared by Boolean specifications. Additionally, the test case generation based on Z language is more mature. But these test cases are in too high degrees of formalizationand in weak commonality, which are difficult to understand and use. At home, interlocking software test case is for the standard station manually designed by experienced debugging engineers, has strong dependence on expertise of testers, which will affect the sufficiency and effectiveness of test. Due to the complexity of security software and the fairness of interlocking software test, the third party usually applies black box testing. UML (Unified Modeling Language) is a standard modeling language specific to the system analysis and design of subject, including a series of views and models, which describes the structure, behavior and use of software from the different levels and angels.Therefore, the software testing method basing on UML model has been always concerned by researchers, and is widely used in development and testing of object-oriented software.

UML Priority Sequence Diagram Model

UML Sequence diagram is a form of interaction diagrams and used to describe the dynamic interaction among objects, which has good interactive expression of software object, and pays much attention to the time sequence of interactive information. In diagram, there are two axes in sequence diagram. Of which, the horizontal axis represents different objects involved in the interaction, and the vertical axis represents time. The object in sequence diagram is expressed by the vertical line with rectangular block. The vertical line is named as the lifeline of object, representing the life cycle of this object in the interaction among objects. The communication information among objects is represented by the arrow among the lifeline of objects. Message nameand some control informationare labeled on the top of arrow. The arrow is divided into solid arrow and dotted arrow. Solid arrow indicates a triggered practical method, and dotted arrow only indicates the returning of one method.

For accurately describe the static interaction process in sequence diagram, and eliminates the software defect caused by ambiguity of natural language, in this paper, the UML sequence diagram is represented with a quad SD = < O, M, L, F> thereof:

O={O1,O2,…,Om}， is the aggregation of objects in sequence diagram.

M={Message|Message=guard_condition*message_name*parameter} is the aggregation of information in sequence diagram. Each message in sequence diagram could be

Expressed as : [monitoring condition]message name（parameter）.

L is the aggregation of site in sequence diagram. Site is the point of sending or receiving messages on the lifeline of objects. Every sites are in the form like this: <Oi,li>， of which, Oi ∈O li is the site of object Oi.

F:M×{s, r}→L is a function relationship from message to site in sequence diagram. Srepresents sending message, r is receiving message. Figure 1 is a simple sequence diagram, the formalized definition of which is showed in figure 2.

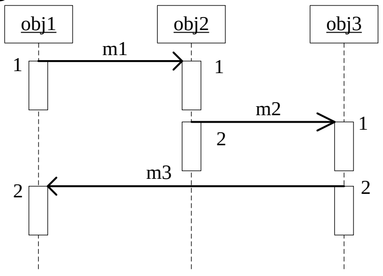

Figure 1 One simple sequence diagram

SD=<{obj1,obj2,obj3},{m1,m2,m3},{<obj1,1>,<obj2,1>,<obj2,2>,<obj3,1>,<obj3,2>,<obj1,2>},{F(m1,s)=<obj1,1>,F(m1,r)=<obj2,1>,F(m2,s)=<obj2,2>,F(m2,r)=<obj3,1>,F(m3,s)=<obj3,2>,<F(m3,r)=<obj1,2>}>

Figure 2 Formalized definition of sequence diagram

Priority generation method of test cases based onUML Sequence Diagram

The test case generation process based on sequence diagram is showed as Figure 3.

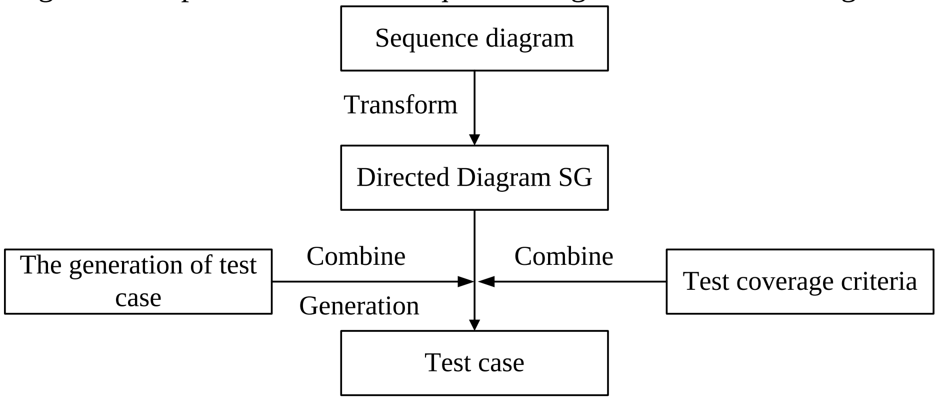

Figure 3 Test case generation process based on sequence diagram

The generation step of test case based on sequence diagram is as below:

(1) Adding constraints to the sequence diagram.

(2)By traversal of all effective sequence of events in sequence diagram, the scenery fits all sequence diagrams is obtained and directed diagram SG is generated.

(3) Defining test coverage criteria and the generation method of test case to generate test case.

The method for generating sequence diagram to directed diagram SG.

Definition1 directed diagramSG=<NSG,∑SG, q0SG,FSG >, thereof:

NSG={N1,N2,…,Nm} is the aggregation of message node, Nj=<messageName,fromObject,toObject,[c()]>， Nj is the node set according to message m. Thereof, fromObjectis the object for sending message, toObject is the object for receiving message, c() is monitoring condition of message; ∑SG is the aggregation of edges of transform among nodes; q0SG is the initial nodes of directed diagram SG, also is the only ingress node; FSG is the aggregation of endpoint of directed diagram SG.

Definition 2 Test scenarioTS:TS of sequence diagram={TS1,TS2,…,TSn}, which represents the path of scenario in sequence diagram including normal scenarioand abnormal scene, thereof:

TSi:<ScnId,StartState,NodeSet,NextState>，thereof, ScnId indicates the single identification No. of testing scene, StartState indicates the initial status of system before performingTS, NodeSet shows the aggregation of message nodes occurred in the testing scenarioTSi.

NextState indicates the status of the system after performing TSi.

The step for transforming UML sequence diagram to directed diagram SG is shown below:

(1)Initialization: NSG=NULL；

(2)Combing the sequence of sending message and constraints in sequence diagram, it generates corresponding message nodes N and the transform edge∑ among nodes.;

(3)Write the active message of nodes with in-degree 0 down as q0, write the postposition of nodes with in-degree 0 down asF.

One directed diagram SDG is only correspond to one starting point, but it may have one or more ending point as to different testing scenario.

Testing coverage criteria

Testing coverage criteria is the measurement to the sufficiency of software test, and is the evidence of validity and reliability of testing result. Any testing shall satisfy certain of coverage rate. Starting from the possibly existing operation mistake of every object, the mistake of message exchange among objects and the mistake of scenario, this papertraverses directed diagram SG and obtain relevant test case set by applying node coverage criteria, transfer edge coverage criteria, and all path s coverage criteria.

Criteria 1 node coverage criteria: the nodes on SG shall be accessed at least one time.
Criteria2 transfer edge coverage criteria: the transfer edges on SG shall be execute at least one time.
Criteria3 coverage criteriafor logic path s: all the path s on SG shall be executed at least one time.

In view of the implicit operation mistake in system, for generating the test case which satisfies the test coverage criteria, it is necessary to traverse the SG to corresponded sequence diagram, get all the path s from the initial node to the end node, access every path and get corresponded test scenario.

The generation method of test case

In order to generate the test case conforming to test case coverage criteria, we applied depth-first search algorithm to transverse directed diagram SG. The detailed algorithm flow chart is as shown in Figure 4.

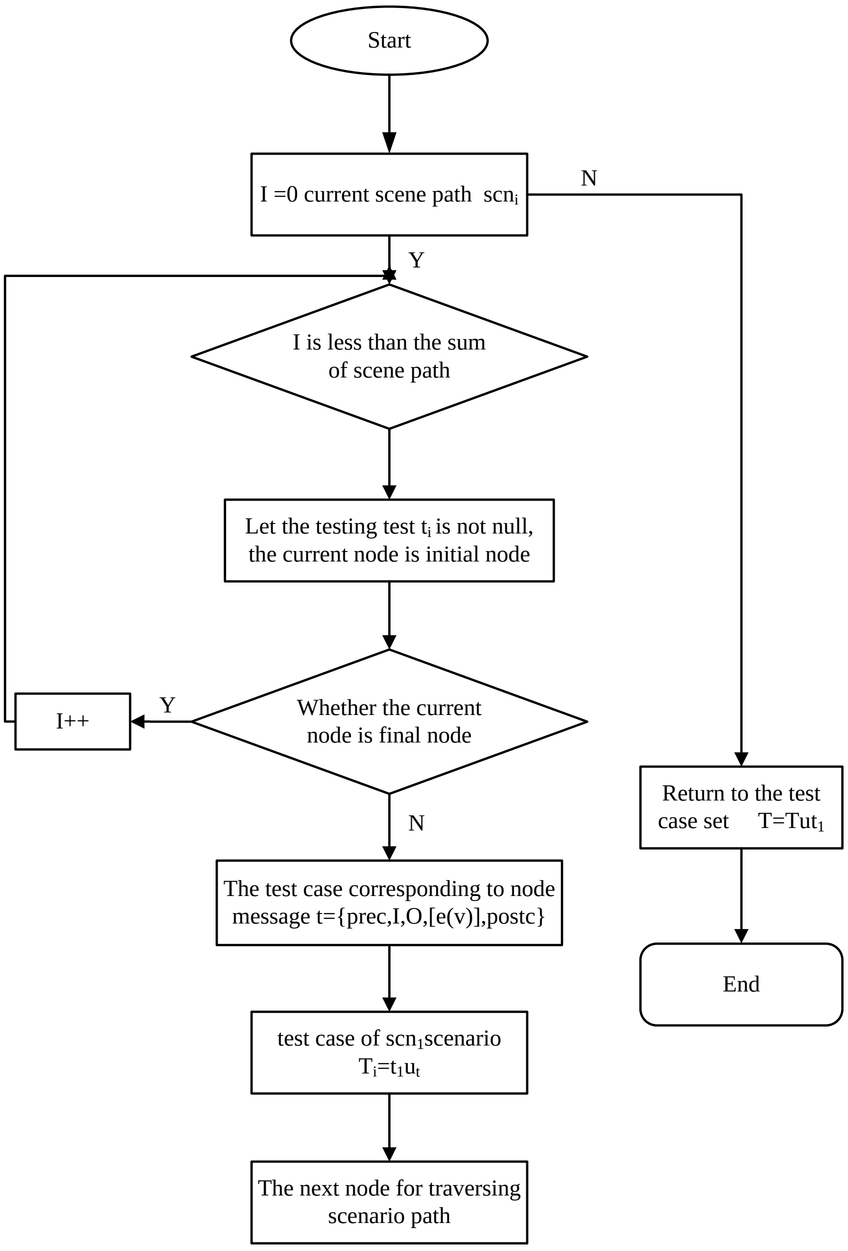
Start
I =0 current scene path scn_i
N
Y
I is less than the sum of scene path
Let the testing test t_i is not null, the current node is initial node
Whether the current node is final node
Y
I++
N
The test case corresponding to node message t={prec,I,O,[e(v)],postc}
test case of scn_1scenario T_i=t_1u_t
The next node for traversing scenario path
Return to the test case set T=Tut_1
End

Figure 4 the generation flow chart of test case

When transverse all directed diagrams SG, keep a record of the constraints included in each nodes including the status message of object, branch of conditions and the input and eventually output of the scene, and the various constraints are linked by the logic and (&). As to random scene, test case is the precondition, input condition, event constraint and the expected output result.

Conclusion

This paper introduces attention requirement coverage rate, the importance degree of test cases basing on requirement and failure rate of test case, to jointly decide the priority matter of test case based on requirement, as well as introduces weight factor aiming to the three factors for guaranteeing engineers to adjust the importance of the three factors in accordance with specific test object and environment, thus to increase the test effectiveness.

Acknowledgement

In this paper, the research was sponsored by the Open Foundation of Innovation Base for Graduate Students of Nanjing University of Aeronautics and Astronautics (ProjectNo. kfjj201460).

Reference

[1]Yishuang Geng, Kaveh Pahlavan, On the Accuracy of RF and Image Processing Based Hybrid Localization for Wireless Capsule Endoscopy, IEEE Wireless Communications and Networking Conference (WCNC), Mar. 2015

[2]Jie He, Yishuang Geng and Kaveh Pahlavan, Toward Accurate Human Tracking: Modelling Time-of-Arrival for Wireless Wearable Sensors in Multipath Environment, IEEE Sensor Journal, 14(11), 3996-4006, Nov. 2014

[3]Lv, Zhihan, Liangbing Feng, Haibo Li, and Shengzhong Feng. "Hand-free motion interaction on Google Glass." In SIGGRAPH Asia 2014 Mobile Graphics and Interactive Applications, p. 21. ACM, 2014.

[4]Zhong, Chen, Stefan Müller Arisona, Xianfeng Huang, Michael Batty, and Gerhard Schmitt. "Detecting the dynamics of urban structure through spatial network analysis." International Journal of Geographical Information Science 28, no. 11 (2014): 2178-2199.

[5]Li, Wubin, Johan Tordsson, and Erik Elmroth. "An aspect-oriented approach to consistency-preserving caching and compression of web service response messages." In Web Services (ICWS), 2010 IEEE International Conference on, pp. 526-533. IEEE, 2010.