A comparative study of data flow testing techniques

Swati Sarraf
Student, M.Tech.(IT)
Amity School of Engineering and Technology
Amity University, Noida, U.P. India

Farah DeebaHasan
Student, M.Tech.(IT)
Amity School of Engineering and Technology
Amity University, Noida, U.P. India

Naveen Garg
Assistant Professor
Amity School of Engineering and Technology
Amity University, Noida, U.P. India

Abstract - This paper describes various methodologies of performingdata flow testing. Since testing is a process of executing a program with the intent of finding an error, so in this paper steps to do data flow testing are describe and how test suites are designed keeping in mind anomalies. This paper gives a idea about types of testing i.e. static and dynamic testing. Further this paper describes the techniques implemented till now to do data flow testing techniques. These techniques include Design by Contract, A Data Flow Coverage, Testing of Web Applications and a comparative study of them.

Keywords : Design by contract,Data flow coverage,web services

I. INTRODUCTION

Testing of a software system is not the responsibility of a single person but it is a team work and the size of team depends on the complexity,criticality and functionality of software under test[5]. To ensure effecting testing test plans and test suites are develop with the start of the software development process.

Since testingcan be manual, automated or the combination of both. In manual testing tester test the program manually and try to find bug while in automated testing a tool is required to test the software. As 100% automated testing is not possible so manual and automated testing are done in combination to test software. When it comes to type of testing then there are two types static and dynamic testing.In static testing the actual source code analysed without executing it. While in dynamic code is executed to find all possible bugs. This paper present survey on data flow testing which is a type of dynamic data flow testing

Section II of this paper gives the basic idea how data flow testing is done? What are all the steps to do data flow testing. What is the need to do data flow testing. Section III describe about the data flow testing by the method of Design by Contract. Section IV present about data flow testing through the method of data flow Coverage. And section V testing of web application through data flow testing method is done. Section VI gives a comparative study about all methodologies.

II. DATA FLOW TESTING

Data flow testing is a refinement of control flow testing technique. Data flow testing has no relation with data flow diagram. It is based on observing the variable and their effect on program execution. It examine the lifecycle of a variable i.e. its definition, its usage in the program, its computation and killing of that variable. Data flow testing is itself of static and dynamic type.

In static data flow testing, testing is done by examining the program without executing the code. When the code is examined then the variable is defined in the program, and then where the usage of that variable occurs in that program. If that variable is been used in the program after it has been killed is a anomaly.

Some of the anomalies are:[5]

i) Variable is defined but never used.
ii) Variable is used but never defined.
iii) Variable is defined twice before it is used.
iv) Variable is used before even first-definition.

These are some of the anomalies which are examined by data flow testing and they help in designing a good test suite for a program.

Definitions used in data flow testing are as follows:

a) **Defining nodes, DEF (v, n)**: A node in a program Q is a defining node for a variable v, only if,at n, v is defined. For example- a node y is a defining node if it states "int a=0;".
b) **Usage nodes,USE(v,n)**: A node in a program Q is a usage node of a variable v only if at n, v is used. For example A node y is a usage node if it states "b=4+a;".
c) **Du-Path**: a path in the set of all paths in Q(G) is a du-path for some variable v only if there exist DEF(v,n) and USE (v,n) nodes such that m is the first node of the path and n is the last node.
d) **Du-Path**: A path in the set of all paths in Q(G) is the dc-path for a variable v only if it is a du-path and the initial node of the path is the only defining node of v in the path.

Static data flow testing fails when the state of a variable can't be determined by just examining the code. This possibility occurs when variable is used as index for a collection of variable element. For example- In array, the index may be generated dynamically during program execution therefore which state of the array element is being referred can't be guaranteed. Static data flow testing may be denoting certain piece of code to be anomalous which never be executing and therefore not anomalous completely.Therefore dynamic data flow testing is neededsome time to check the bug during the execution of the program.

The following flow chart describe the step to do data flow testing.

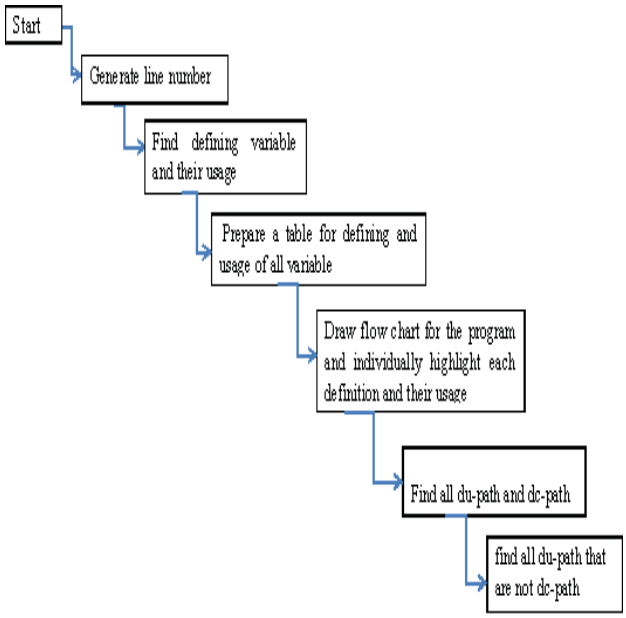

Figure 1 : Flow Chart for Data Flow Testing

III. Enhancing data flow testing through design by contract

DBC i.e. Design By Contract is a methodology to construct quality software. Dbc uses contract specification means pre and postconditions of methods of class and class invariant with the design of a class. In this methodology data flow testing and design by contract is combined to develop a testing technique called as "Data flow testing using Design By Contract". In this technique from the contract specification class flow graph is generated and then conventional data flow testing is applied to find test cases.

The advantage of using contract specification is that it is executable and hence a method in a test case is only executed if its precondition and class invariant are satisfied, if not satisfied it will result into infeasible sequence and also if after executing a method its postcondition is not satisfied then it leads to an implementation error in the method. A contract specification is made of following components:

1.) **Precondition of method**: It defines the conditions which should be true before the execution of the method of the class. Each method of the class has a precondition.
2.) **Postcondition of method:** It defines the conditions which should be true after the execution of the method of the class. Each method of the class has a postcondition
3.) **Class Invariant:** Class invariant states the conditions which should be true for all the objects of a class. The invariant must be satisfied after the execution of a method if it was satisfied before the execution of the method. Invariant is a constraint that checks consistency of the state of an object throughout its life cycle. Invariant should be true after the creation of every object of a class

Data flow testing using Contract:A class implementation is tested with respect to the contract specification of the class. The technique consists of following steps:

1.) Given a class constructed through Dbc, a class flow graph is generated form its contract specification. The contract specification includes preconditions and post-conditions for the methods of the class and invariant for the class.
2.) The definitions and uses(p-uses and c-use) of each data member of the class are identified and infeasible du pairs are eliminated based on the contract specification.
3.) The test cases are generated from CFG using conventional data flow testing criteria.

A class flow graph(CFG) is a directed graph G=(N,E) whereN is set of nodes. Noderepresents the methods of the class. E is set of edges.Method nodes are connected by edges.Contract specification of the class decides whether an edge exists between two method nodes or not.An edge is drawn from method node let say M1 to M2 if the post conditions of M1 satisfies the precondition of M2 and invariant of the class is true. Each data member of a class is classified as being defined,computation used(c-u) or predicate used(p-use) and this is identified by the following rule: Rule: Let G be a CFG of a class C and d be its data member.

a) d is defined at method node if this method assigns a value to d.
b) d is said c-used at method node if this method references d.
c) d is said p-used at method node if it is used in a condition in the method.

so the conclusion is, a technique for class testing using design by contract and data flow testing is developed,to test the class data flow testing criteria is used. Although there are other class testing technique,which uses flow graph and apply data flow testing criteria to generate test cases but this technique is different from other technique in the following way:

1. This technique uses flow graph which is constructed from contract based specification.
2. Infeasible sequences are eliminated.
3. Implementation level errors can be found.

IV. DFC(Data Flow Coverage)

DFC is the tool implemented at the Institute of Computer Science Warsaw University of Technology as an Eclipse plugin. The objective is to present dataflow testing of java programs supported by DFC.It finds all definition uses pair in tested unit and also provides the definition uses graph for methods.

DFC – a tool for dataflow testing: Data flow testing can not be applied in isolation so to support this approach a tool is implemented i.e. DFC (Data Flow Coverage) as an eclipse plug-in. The java programming environment and many testing tools like JUnit is available in Eclipse.DFC finds all def-u pairs in testing java code and after test provides the tester info. which def-u pairs were covered? On the basis of this information the tester can decide which coverage criteria should be used and add appropriate test cases. For the test case preparation the tester can also use def-use graph for a method which is provided by DFC

Figure 2 is used to represent the idea for testing using the DFC[3]

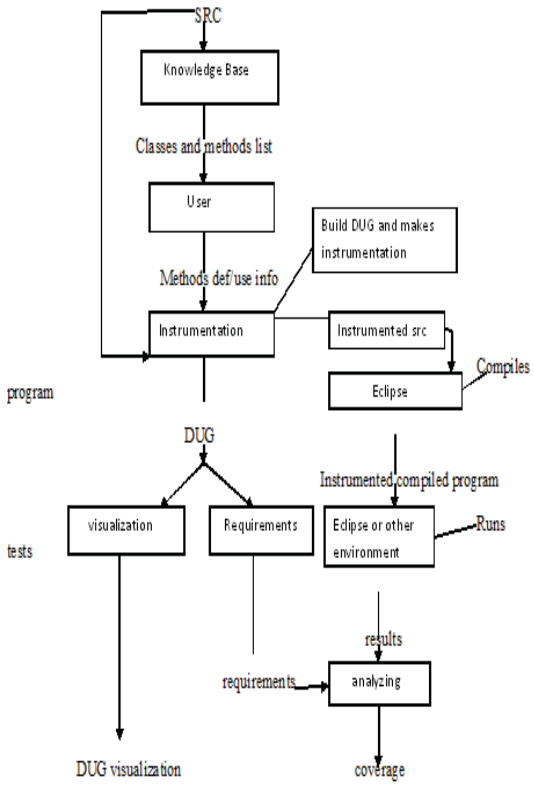

Figure 2 : Data flow testing via DFC

In the figure the main parts of DFC and its collaboration with the Eclipse environment are presented. Java source code(SRC) is the input for DFC, source code analyzation is done by module knowledge base and it generate the list of classes along with methods. On the list tester marks the methods as modifying or using object state. The module instrumentation adds extra instructions needed for finding dataflow coverage and builds def-use graph(DUG) which contain information concerning control flow, variable definitions and usage in its node. DUG is the input for module visualization, drawing the graph and Requirements-finding all def-u pairs. The instrumented code should be compiled and run in Eclipse environment. Instrumentation module adds some extra code which sends data concerning coverage to DFC. Module analysing is locating covered and not covered def-u pairs.Details on DFC implementation and its usage can be found in[8]

So the conclusion is that by supporting data flow testing of java classes, opportunities to find errors that may not be uncovered by black-box testing is provided. In Eclipse environment there are other tools available for testing

java programs using different techniques eg. JUnit , EcEmma or TPTP. In DFC tester can design tests to achieve for eg.. Def-uses or all-uses coverage criteria which also guaranteeinstruction coverage.

V. TOOL FOR WEB SERVICES BASED ON DATA-FLOW

For large and complex composite web services which are composed by BPEL and a set of WSDL and XSD documents, there is requirement of tools in designing and selecting test cases. The theory of data flow testing is applied and a testing tool for web services composition is developed which can automatically detect harmful data flow anomalies and generate test paths according to all kinds of data flow coverage criteria.

The architecture of this testing tool is:[4]

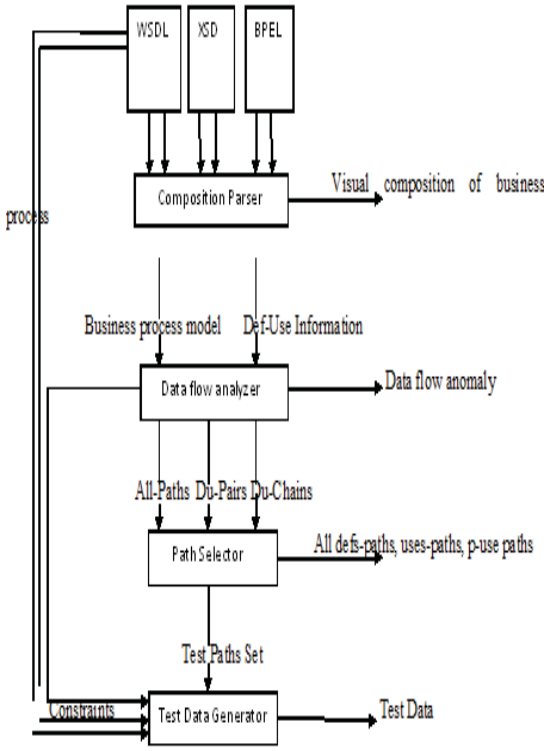

Figure 3 : Web services Testing Tool

Composition parser: From WSDL and the XSD documents the variable is parsed layer by layer and all the atom variables are extracted, then information of the variable is recorded in basic activity nodes in the process of BPEL document to the control flow graph. The output is the visualization of process where control flow and data information are accurately specified .It help testers verify whether the process corresponding to BPEL code is consistent with the design intent.

Data flow analyser: It finds definitions, p-uses, du-chains, du-pairs, data flow anomalies and preserves them in each generated execution path through search of the model

Path Selector: A set of test paths for each data flow criteria is easily generated as the covered def-use information corresponding to all kinds of data flow criteria has been preserved in each path.

Test Data generator: Based on the type and even bound of some variables defined in WSDL and related XSD documents and the combine analysis of constraints saved in switch node,part of test data can be generated from the selected paths.

Hence this tool support the feature of BPEL and automatically select the test path according to a series of data flow testing strategies and without execution of process the main patterns of incorrect data used can be recognized.

VI. CONCLUSION

This paper discuss about various methodologies for implementation of data flow testing So following table depicts the comparative study of the three methodologies implemented so far:

Table 1 : Comparison Between Various Methodologies for Data flow Testing

Techniques / parameters	Design by contract	Data flow coverage	Web services
Basic approach	Contract specification(post and pre condition, class invariant)	Def-u pairs generated based on the intra-method approach	Support BPEL features and automatically select test path acc. To series of data flow strategies
Advantages	-Implementation level error can be found. -infeasible sequence are eliminated.	Addition of test cases after execution by the tester.	Main patterns of incorrect data can be recognized without execution of process
Application area	coinBox [2][7]	Bill generation[3].	Train service, airline service hotel agent service etc.

REFERENCES

[1] K. K. Aggarwal and Yogesh Singh,: Software Engineering. In: New Age International Publishers.
[2] Yogesh Singh,AnjuSaha "Enhancing data flow testing of classes through Design by Contract", Seventh IEEE/ACIS International Conference on Computer and Information Science,2008
[3] IlonaBluemke, ArturRembiszewski,"Dataflow approach to testing Java Programs", fourth international conference on dependability of computer systems,2009
[4] Jun Hou, Lei Xu,"A Testing Tool for Composite Web Services based on data flow",Sixth web information systems and applications conference,2009
[5] yogeshsingh, "software testing" Cambridge university press,2012,pp 173-175.
[6] Boris Beizer, "software testing techniques", International Thomas Computer Press,1990
[7] Kung D & Lu,N Venugopalan &N , Hsia & p., Toshiyama & Chen, C & Gao, J, "Object state testing and fault analysis for reliable software systems, 7th International symposium on Software reliability engineering, New York,pp 76-85 ,1996
[8] A.Rembiszewski "Data flow coverage of object programs" Msc thesis,Institute of Computer Science,Warsaw university of technology,2009